Fuzzy and possibilistic co-clustering techniques for high-dimensional data analysis

Tjhi William Chandra

2008

Tjhi, W. C. (2008). Fuzzy and possibilistic co-clustering techniques for high-dimensional data analysis. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/3503>

<https://doi.org/10.32657/10356/3503>

Nanyang Technological University

Downloaded on 26 Apr 2023 11:42:43 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**FUZZY AND POSSIBILISTIC CO-CLUSTERING
TECHNIQUES FOR HIGH-DIMENSIONAL DATA
ANALYSIS**

**TJHI WILLIAM CHANDRA
SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING
2008**

Fuzzy and Possibilistic Co-clustering Techniques for High-dimensional Data Analysis

Tjhi William Chandra

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2008

Acknowledgements

Thanks be to God, the source of all knowledge and inspirations.

I would like to express my utmost gratitude to my supervisor, Dr. Chen Lihui, for the constant guidance, the words of encouragement, the frequent long-hour meetings, the working weekends, and the invaluable training throughout this research project.

I would like to thank the special people in my life: my brother who stays with me through the joys and the pains; my girlfriend for the love, encouragement, and prayers; and my parents for their advices and constant support.

I would also like to take this opportunity to thank Erlin, Kai Cheng, Zhang Zhang, Zhang Xin, Minh Duc, Yao Wei, and all the other FYP students who have helped me in the project.

Special thanks also to all the very helpful Software Engineering Lab. technicians: Poh Huat, Ms. Hao, Christina, Annie, and Francis.

Finally, I would like to express my sincere gratitude to Dr. Chiam Tee Chye, who made this research project possible in the first place.

Summary

In this thesis, we present our study of some new data clustering mechanisms for high-dimensional data analysis. Our objective is to simultaneously achieve several goals for data analysis namely: effective clustering in high dimension, rich and natural representations of clusters, robustness to outliers, and highly-interpretable clusters. Novel clustering models based on the fuzzy and possibilistic co-clustering frameworks have been developed for the purpose.

Co-clustering, a technique of simultaneous clustering of objects and features, is regarded by many in the literature to be one of the most effective approaches to automatically categorize high-dimensional data. Fuzzy co-clustering enhances standard crisp co-clustering by capturing a more realistic fuzzy representation of *co-clusters*, i.e. the resulting associated object and feature clusters. Existing prominent fuzzy co-clustering algorithms however are vulnerable to outliers. In addition, the common mathematical model that underlies these existing algorithms can be shown to be too rigid in formulation. This rigidity prevents possible expansions of fuzzy co-clustering that can potentially address the outlier problem, as well as enrich the representations of co-clusters.

We introduce a new fundamental and less rigid formulation for fuzzy co-clustering based on the dual-partitioning approach. In essence, the dual-partitioning approach is like the Fuzzy C-means in the context of fuzzy co-clustering, and it offers a solid

foundation for further expansions of fuzzy co-clustering. We propose three different dual-partitioning-based fuzzy co-clustering models. Our analyses and experiments on large document datasets conclude that two of the models, Improved Partitioning-based Fuzzy Co-clustering (IPFC) and Heuristic Fuzzy Co-clustering with Ruspini’s condition (HFCR), are effective for practical clustering tasks. Further analysis also reveals that between the two, IPFC provides a more solid theoretical formulation than the heuristic-based HFCR, while HFCR offers a relatively simpler fuzzy co-clustering than IPFC.

Accordingly, we propose two expansions based respectively on IPFC and HFCR to achieve our goals of improving co-clustering’s robustness to outliers and making the corresponding co-cluster representations richer and more natural. In these expansions, we introduce possibilistic memberships into fuzzy co-clustering, resulting in a new powerful framework called fuzzy-possibilistic co-clustering. Possibilistic clustering is known in the literature to offer robustness to outliers and highly realistic cluster membership based on the typicality measure. The two proposed fuzzy-possibilistic co-clustering models are Dual Fuzzy-possibilistic Co-Clustering (DFPC), which is based on IPFC; and Possibilistic-Fuzzy Co-Clustering (PFCC), which is based on HFCR. DFPC offers a general-purpose fuzzy-possibilistic co-clustering that has a slower real-time computation than PFCC; while PFCC offers fuzzy-possibilistic co-clustering that is more efficient than DFPC in real-time, but with limited applications for large datasets. Aside from these trade-offs, with the incorporation of possibilistic memberships, both models can offer improved robustness to outliers, while simultaneously maintain the effectiveness in clustering high-dimensional data. Furthermore, with their combined fuzzy and possibilistic memberships, DFPC and PFCC allow more natural and richer clusters representations than their fuzzy co-clustering counterparts. Our analyses and experiments on artificial and large benchmark document datasets confirm all these claims on the advantages of fuzzy-possibilistic co-clustering.

We also investigate one important practical application of fuzzy-possibilistic co-clustering, namely the automatic generation of descriptive document clusters' summaries. Literature agrees that, in the context of document-word co-clustering, the word clusters produced by co-clustering algorithms can be used to capture the contents of the associated document clusters. By utilizing an existing multi-document summarizer, we apply the robust and realistic soft word cluster memberships of DFPC and PFCC to generate description summaries of document clusters. Based on our simulation results, we find that the incorporation of the soft word cluster memberships improve the interpretability of the document summaries.

Contents

Acknowledgements	i
Summary	ii
Contents	v
List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Overview	1
1.2 Motivations	3
1.3 Objective	5
1.4 Contributions	6
1.5 Organization	8
2 Research Backgrounds	9
2.1 Overview	9
2.2 Literature Review	11
2.2.1 Hierarchical Clustering	11

2.2.2	Partitional Clustering	13
2.2.3	Density-based Clustering	14
2.2.4	Fuzzy Clustering	15
2.2.5	Graph-based Clustering	16
2.2.6	Model-based Clustering	17
2.2.7	Clustering High-dimensional Data	19
2.2.8	Why Fuzzy-possibilistic Co-clustering	24
2.3	Relevant Works	25
2.3.1	The FCM Model	26
2.3.2	The PCM Model	28
2.3.3	The PFCM Model	31
2.3.4	The FCCM Model	32
2.4	Conclusions	36
3	The Dual-partitioning Approach to Fuzzy Co-clustering	38
3.1	Overview	38
3.2	The Definition of the Dual-partitioning Approach	39
3.3	The Limitation of the FCCM Model	40
3.4	The Proposed Models for the Dual-partitioning Approach	44
3.4.1	The First Model: The FCR Algorithm	44
3.4.2	The Second Model: The HFCR Algorithm	54
3.4.3	The Third Model: The IPFC Algorithm	59
3.5	Advantages and Disadvantages of the Dual-partitioning Approach	66
3.5.1	Advantages	68

3.5.2	Disadvantages	72
3.6	Experimental Results	74
3.6.1	Implementation Details	74
3.6.2	Results and Discussions	78
3.7	Conclusions	83
4	The Dual Fuzzy-possibilistic Co-clustering	85
4.1	Overview	85
4.2	The Motivation	86
4.3	The Proposed Model	88
4.3.1	The DFPC Algorithm	88
4.3.2	The Proof of Convergence	94
4.4	Experimental Results	98
4.4.1	Robustness to Outliers	98
4.4.2	Document Clustering	110
4.5	Conclusions	118
5	The Possibilistic-fuzzy Co-clustering	121
5.1	Overview	121
5.2	The PFCC Model	122
5.2.1	The Objective Function	122
5.2.2	The Update Equations	126
5.2.3	The Pseudo-code	129
5.2.4	The Proof of Convergence	130
5.3	PFCC and DFPC: An Analytical Comparison	133
5.4	Experimental Results	135
5.4.1	Robustness to Outliers	135

5.4.2	Document Clustering	145
5.5	Conclusions	157
6	Summarization of Document Clusters	158
6.1	Overview	158
6.2	Literature Review	159
6.2.1	Multi-document Summarization Review	160
6.2.2	The MEAD Summarizer	161
6.2.3	ROUGE Evaluation Measure	163
6.3	Word Membership-based MDS	164
6.3.1	The Motivation	165
6.3.2	The Word Membership-based MEAD	166
6.3.3	Experiments	167
6.4	Conclusions	173
7	Conclusions and Future Works	176
7.1	Conclusions	176
7.2	Future Works	182
	Author's Publication List	184
	Bibliography	186

List of Figures

2.1	Standard Clustering vs. Co-clustering	22
2.2	Outlier Problem	29
3.1	An Example D_1	51
3.2	FCR Simplified Memberships Search Spaces	52
3.3	A Common Clustering Situation 1	67
3.4	A Common Clustering Situation 2	69
3.5	An Example D_2	71
3.6	Performance Comparison of FCR, HFCR, and IPFC	82
4.1	2-D Dataset with 23 Points	99
4.2	2-D dataset with 23 points labeled based on u_{ci}	103
4.3	2-D dataset with 23 points labeled based on x_{ci}	104
4.4	Outlier-contaminated Dataset D_{o1}	106
4.5	Performance Comparison of DFPC, IPFC, and Fuzzy CoDoK	115
4.6	Bottom 4 <i>Classic3</i> Documents with the lowest DPFC's $\sum_{c=1}^C x_{ci}$ values	116
4.7	Common <i>Classic3</i> Documents	117
5.1	2-D Dataset with 602 Points	137
5.2	2-D dataset with 602 points labeled based on u_{ci}	139
5.3	2-D dataset with 602 points labeled based on x_{ci}	139

5.4	Outlier-contaminated Dataset D_{o2}	141
5.5	Performance Comparison of PFCC, HFCR, and Fuzzy CoDoK	150
5.6	Performance Comparison of PFCC and DFPC	151
5.7	Bottom 4 <i>Classic3</i> Documents with the lowest PFCC's $\sum_{c=1}^C x_{ci}$ values .	152
5.8	Common <i>Classic3</i> Documents	153
6.1	<i>Classic3</i> Document Clusters Summaries	167
6.2	Summarization ROUGE Performance Comparison	172
7.1	The Contributions of this Thesis	180

List of Tables

2.1	List of Mathematical Notations	26
3.1	List of Mathematical Notations	40
3.2	FCR Pseudo-code	47
3.3	HFCR Pseudo-code	56
3.4	IPFC Pseudo-code	63
3.5	List of Document Datasets	75
3.6	Document Clustering Parameters Setting	77
3.7	Document Clustering Performance	80
4.1	List of Mathematical Notations	88
4.2	DFPC Pseudo-code	94
4.3	2-D Dataset with 23 Points	100
4.4	The Memberships of the 2-D Dataset with 23 Points	102
4.5	Parameter Settings for the 2-D Experiments with 23 Points	105
4.6	The Average Object Memberships of D_{o1}	108
4.7	The Average Feature Memberships of D_{o1}	108
4.8	Document Clustering Parameters Setting	112
4.9	Document Clustering Performance	114
4.10	DFPC Possibilistic Memberships of <i>Classic3</i> 's doc_1 - doc_5	117

4.11	DFPC Partitioning Memberships of <i>Classic3</i> 's <i>doc</i> ₁ - <i>doc</i> ₅	117
4.12	<i>Multi5</i> 's Top Words	119
5.1	List of Mathematical Notations	123
5.2	PFCC Pseudo-code	130
5.3	Real-time Computations Comparison: PFCC's v_{cj} vs. DFPC's v_{cj}	134
5.4	Real-time Computations Comparison: PFCC's y_{cj} vs. DFPC's y_{cj}	135
5.5	Real-time Computations Comparison: PFCC's x_{ci} vs. DFPC's x_{ci}	136
5.6	The Average Memberships of the 2-D Dataset with 602 Points	138
5.7	Parameter Settings for the 2-D Experiments with 602 Points	140
5.8	The Object Memberships of D_{o2}	143
5.9	The Feature Memberships of D_{o2}	143
5.10	Document Clustering Parameters Setting	146
5.11	Document Clustering Performance	149
5.12	Comparison of Real-time Performance per Iteration	151
5.13	PFCC Possibilistic Memberships of <i>Classic3</i> 's <i>doc</i> ₁ - <i>doc</i> ₅	153
5.14	HFCR Partitioning Memberships of <i>Classic3</i> 's <i>doc</i> ₁ - <i>doc</i> ₅	153
5.15	<i>Multi5</i> 's Top Words	154
5.16	Validity of Assumption 1	156
6.1	DPFC-MEAD Summarization Performance on DUC 2004	173
6.2	PFCC-MEAD Summarization Performance on DUC 2004	173
6.3	FC-MEAD Summarization Performance on DUC 2004	174
6.4	MEAD Summarization Performance on DUC 2004	174
7.1	Practical Implications and Applications of Fuzzy-possibilistic Co-clustering	181

Chapter 1

Introduction

1.1 Overview

We are now living in the information age. The surface Web alone contains approximately 19.2 billion web pages, according a study published by Yahoo! in August 2005. While having a lot of information is generally a positive thing, the huge sizes of data repositories that we have now pose a challenge for us to be able to find relevant information effectively and efficiently. Because of this, and the fact that most real-world datasets are high-dimensional in nature, there is a pressing need for tools that can lead to an automation of large and high-dimensional dataset analysis. One technology that can help to meet this demand is clustering of high-dimensional data. And this is the core topic of this thesis.

Data clustering is defined as a process of automatic categorization to discover natural groupings of similar objects in a dataset. The ability to automatically categorize patterns is generally considered to be one of the most fundamental functions of the human cognition system [136]. We can, for example, sense two clusters of writing themes after reading dozens of computer and sport articles. If this ability can be effectively

emulated in machines, it will likely cause a significant progress in our technology.

Today data clustering has been applied in several fields, usually for the purpose of data organization or assistance in decision-making processes. Some applications of clustering include:

- Image segmentation for examinations of medical images, handwriting recognition, and analysis of photos from satellites [54, 63].
- In information retrieval, clustering has been used for automatic document categorization, summarization of documents based on sentence clustering, and query expansion to retrieve documents [107].
- In business, customers can be clustered according to their behaviors [134]. Additionally, clustering has also been applied for market segmentation and product categorization [18].
- In bioinformatics, we have clustering of DNA microarray data [69]. Clustering has also been used to assign classes of nonannotated genes based on the annotated ones, and to help in cancer diagnosis [88].
- In the Web, websites like <http://www.clusty.com>, <http://www.webclust.com>, and <http://www.mooter.com> offer services to cluster search results into more comprehensible categories. Clustering of social networks is also another popular clustering application in the Web [133].

The above applications are yet to utilize the full potential of clustering, that is, either they still require human intervention or their performances can still be improved. A study of clustering, particularly for high-dimensional data, therefore remains an important topic in knowledge discovery, considering the potential benefits clustering can bring.

The remaining of this chapter is organized as follows. In Section 1.2, we elaborate the motivations of this study. Section 1.3 explains our objective. In Section 1.4, we highlight the contributions of this thesis. Finally, Section 1.5 explains the organization of this thesis.

1.2 Motivations

Despite the fact that clustering comes naturally to the human learning process [136], achieving an equally effective clustering on machines is a very challenging task. For one, what constitutes clusters may not always be obvious. Even if they are obvious in one context, the same clustering may not be optimum in other different contexts. Some other main challenges to achieve an effective clustering engine are discussed as follows:

- *The complex nature of real-world datasets.* Clustering algorithms in general can effectively categorize well-separated objects in 2-D or 3-D space. In reality however, most real-world datasets, such as web documents and gene expression data, are high-dimensional, or containing many overlaps, or both. In a high-dimensional space, the curse of dimensionality makes it difficult to find cluster patterns [117]. Overlap among categories also makes it difficult for a clustering algorithm to find separation boundaries among clusters, which are usually non-linear. In addition to all these, there is also an increasing demand for clustering algorithms that can handle mixed media types, e.g. the combination of textual and video contents from the YouTube website.
- *Noise and outliers.* In real-world datasets, the presence of noise and outliers is generally unavoidable. Noise is unwanted information that may have come from inaccurate contents or missing attributes [53]. Outliers are data that are

significantly different from the rests of the dataset that it becomes unnatural to have them considered in the dataset analysis. The Web, for example, is heavily polluted by noise and outliers because of its open nature. Clustering algorithms that do not have a mechanism to identify noise and outliers may not be able to perform well in real-world environments.

- *Interpretability of clusters.* Clusters of a simple 2-D or 3-D dataset can usually be easily interpreted by human beings. However, when the complexity of datasets increases, clustering results become more difficult to interpret, even for human experts. For example, clusters of target customers do not tell much if they come without any description of the characteristics of each customer cluster. In the knowledge discovery system, the likely scenario is that clustering outcomes are to be used as inputs to other decision-making modules. Unless clustering algorithms have an integral mechanism to generate interpretable results, at least to the other modules that utilize the resulting clusters as their inputs, a well-automated knowledge discovery system may not be truly realized.
- *Parameter settings.* We mentioned earlier that what is perceived to be optimum clustering can vary from one situation to the others. One mechanism that is often adopted by many algorithms to cater for this change in environments is by having some parameters that can be adjusted. Hence, ideally for every given situation, there is a set of optimum parameter values that can direct the algorithm to discover the optimum clusters. Parameters also often act as the required inputs for some algorithms, like the number of clusters parameter in K-means and Fuzzy C-means [85, 9]. The problem is that most clustering algorithms, instead of being able to automatically decide the correct parameter settings, demand users to provide these parameter values. In many cases, this limitation prevents clustering

algorithms from being practical enough for real-world applications.

- *Scalability.* On the one hand, clustering process generally involves heavy computations. On the other hands, we are faced with an ever-increasing volume of data. Furthermore, there are many applications, such as clustering of web search results, that demand almost instantaneous clustering outcomes. Therefore for clustering to be practical, it is important for clustering algorithms to maintain low computational complexities, or alternatively to incorporate various techniques to improve scalability such as sampling [68, 100].

There are many other challenges to achieve effective clustering [53]. The gap between these difficult challenges and the potential benefits of clustering, is what motivates us to conduct a study in the field. We hope that our study can bring us a few steps closer to generally effective and practical clustering.

1.3 Objective

The goal of this study is to improve clustering analysis in the simultaneous aspects of: applicability in high-dimensional space, quality of cluster representations, robustness to outliers, and interpretability of clusters. We utilize techniques from several existing fields of study: optimization theory, fuzzy set theory, co-clustering, and possibilistic clustering; resulting in the new proposed framework of soft co-clustering based on the combined fuzzy-possibilistic approach. For simplicity, from this point onward, we simply refer to the new framework as *fuzzy-possibilistic co-clustering*. To be more specific, through the fuzzy-possibilistic co-clustering framework, we can achieve an effective performance for high-dimensional data as well as highly interpretable clustering results due to co-clustering, realistic and rich clustering representations due to the combined fuzzy

and possibilistic formulations; and robustness to outliers due to possibilistic clustering. In addition, our framework is formulated based on a simple yet sound optimization technique to keep the proposed algorithms efficient.

1.4 Contributions

In the process of achieving our objective, we have made several contributions to the field of clustering, as follows:

- A comprehensive review of the field of clustering, capturing the important features of some major approaches to clustering, as well as some of their recent developments (Chapter 2).
- The identification, through a mathematical analysis, of the conceptual limitation of one existing practical fuzzy co-clustering model, which we refer to as the FCCM model [101, 75], that prevents the model’s expandability to more advanced co-clustering formulations (Chapter 3).
- The development of three new methods to address the expandability issue in the FCCM model, resulting correspondingly in three basic fuzzy co-clustering models based on the dual-partitioning approach that can be expanded to more advanced co-clustering formulations (Chapter 3). By dual-partitioning we mean that the models can be considered to be equivalent to a dual-Fuzzy C-means (dual-FCM) [9] on objects and features of datasets. FCM is arguably the most basic form of fuzzy clustering, which serves as the foundation of many more advanced fuzzy clustering algorithms [59, 91]. In a similar manner, the three dual-partitioning-based fuzzy co-clustering models that we introduce can serve as the foundations of the soft co-clustering framework that we intend to build.

- The formulation of a new fuzzy-possibilistic co-clustering model called Dual Fuzzy-possibilistic Co-clustering (DFPC) (Chapter 4). Inspired by the work in [103], in which fuzzy clustering is hybridized with possibilistic clustering, in DFPC, we hybridize fuzzy co-clustering with possibilistic co-clustering. We use one of the dual-partitioning-based fuzzy co-clustering models as the theoretical foundation for DFPC. Incorporation of the possibilistic framework into the formulation brings benefits like robustness to object and feature outliers, as well as richer and more interpretable representations of co-clustering.
- The formulation of a new fuzzy-possibilistic co-clustering model called Possibilistic-fuzzy Co-clustering (PFCC) for large datasets (Chapter 5). PFCC allows a simpler and faster soft co-clustering than DFPC, within a specific domain of application, i.e. categorization of large datasets. Being a fuzzy-possibilistic co-clustering algorithm, except for the trade-off between a simpler formulation and a limited scope of applications, PFCC retains all the benefits of DFPC.
- A study of automatic summarization to generate document cluster descriptions as one possible and promising application of fuzzy-possibilistic co-clustering (Chapter 6). In this study, fuzzy and possibilistic word cluster memberships, which are generated simultaneously with the corresponding document clusters, are incorporated into an existing multi-document summarizer MEAD [106] to generate descriptive summaries of the document clusters. Through some simulations, we show that the resulting enhanced summarizers can achieve improved performances, indicating that the soft word memberships may contain exclusive information that is useful for summarization purpose.

1.5 Organization

The rest of this thesis is organized as follows. In Chapter 2, we provide a literature review of the field of clustering, justify the need for the a new fuzzy-possibilistic co-clustering, and discuss in details some relevant existing works that provide the bases of our proposed framework. In Chapter 3, we build the foundation of our proposed fuzzy-possibilistic co-clustering by first addressing the expandability issue in one existing fuzzy co-clustering framework called the FCCM model [101, 75], and second by building three new models for fuzzy co-clustering based on the dual-partitioning approach [127, 126, 124, 125]. We discuss our first fuzzy-possibilistic co-clustering model called DFPC in Chapter 4 [123]. PFCC, a simpler fuzzy-possibilistic co-clustering model for categorization of large datasets, is introduced in Chapter 5 [128]. Chapter 6 provides a study of automatic summarization to generate document cluster descriptions as one potential application for fuzzy-possibilistic co-clustering [123]. The conclusions of this thesis and some future research directions are given in Chapter 7.

Chapter 2

Research Backgrounds

2.1 Overview

Data clustering is a process of automatic categorization to discover natural groupings of similar objects in a dataset. Mathematically, data clustering can be defined as in Definition 1, assuming we have C clusters. This definition is adapted from [34].

Definition 1 *Given a set of objects $O = \{o_1, o_2, \dots, o_N\}$ and the number of cluster C , clustering can be seen as a problem of a function mapping $f : D \rightarrow \{u_{11}, u_{12}, \dots, u_{CN}\}$, such that u_{ci} is high if $Sim(o_i, c)$ is high, and vice versa.*

In Definition 1, u_{ci} denotes the membership function of object o_i in cluster c , and $Sim(o_i, c)$ denotes the similarity measure between o_i and cluster c . In crisp clustering, u_{ci} is either 1 or 0; while in fuzzy or any other soft clustering, u_{ci} can be equal to any real value, usually between 0 and 1.

Different algorithms capture the similarity measure $Sim(o_i, c)$ differently. Perhaps the most common way is by using popular distance functions such as the Euclidean distance, the Manhattan distance, the Jaccard coefficient, the cosine distance, and the

Pearson correlation [53, 118]. There are also other measures to capture the similarity $Sim(o_i, c)$, including the degree of aggregation [101] (which is used in our proposed algorithms), the Kullback-Leibler (KL) divergence [74], the degree of shared neighborhood [35], the confidence of a frequent object set [52], and the matrix density [89]. The nature of dataset is an important factor should be taken into consideration to decide the appropriate similarity measure. For example, in the case of a low-dimensional dataset with numerical features, the Euclidean distance can be a suitable selection; while in the case of a high-dimensional dataset, the cosine distance is perhaps a better choice. The clustering approach adopted is another important factor which should be taken into account when selecting the similarity measure. For example, in the co-clustering approach (i.e. simultaneous clustering of objects and features), the degree of aggregation, which is a function of object and feature cluster memberships, can be a reasonable choice to measure similarities; while in the probabilistic approach to clustering, the KL divergence, which is a function of probability distributions, can serve as a natural (dis)similarity measure.

While the essence of clustering should remain inline with the Definition 1, multiple aspects of clustering (such as the choice of similarity measures discussed above) result in a very large number of different clustering algorithms available in the literature. In this chapter, we intend to provide a brief review of popular approaches to clustering algorithm in the literature. Through the discussions, we can provide the backgrounds necessary to understand our work. The discussions can also clarify the position of our proposed fuzzy-possibilistic co-clustering algorithms with respect to the collection of already existing clustering algorithms, and at the same time give justifications why we need a new framework.

In Section 2.2, we provide the literature review of clustering. This is followed by some explanations on why we need fuzzy-possibilistic co-clustering. In Section 2.3, we

give more detailed discussions on several existing models that serve as the foundations of our work. Section 2.4 concludes this chapter.

2.2 Literature Review

There are many ways clustering approaches can be categorized. In this section, we adopt a rather conventional categorization, in which clustering is divided into hierarchical clustering, partitional clustering, density-based clustering, fuzzy clustering, graph-based clustering, model-based clustering, and clustering for high-dimensional data. At the end of this section, we explain why we need fuzzy-possibilistic co-clustering among many existing clustering algorithms.

2.2.1 Hierarchical Clustering

Hierarchical clustering generates different levels of clustering arranged in a hierarchical structure. It is particularly useful for certain applications where organization of clusters into hierarchy comes naturally, e.g. in generation of biological taxonomy and in categorization of web search results. There are two major types of hierarchical clustering: hierarchical agglomerative clustering (HAC) and hierarchical divisive clustering (HDC).

Hierarchical Agglomerative Clustering

HAC [68] begins with every object in a dataset being a separate cluster from one another, and then proceeds by recursively merging similar clusters until we end up with a single cluster at the top of the hierarchy. Different ways of measuring similarity during the merging in HAC result in different variants of algorithm. For example, we have a single-linkage algorithm if the minimum distance between two clusters is used, a complete-linkage algorithm if the maximum distance between two clusters is used, and

a centroid-linkage if the distance between two clusters' centroids is used. Depending of the similarity measure used, HAC can result in different clustering outcomes. Studies on the impact of similarity measures on HAC performance have been conducted in [90]. There are several well-known algorithms developed to improve the basic HAC, namely BIRCH [144], CURE [49], ROCK [50], and Chameleon [67]. All these HAC algorithms are informative because they provide clustering information at different levels of hierarchy. However, for the same reason, they can be less efficient compared to other non-hierarchical clustering methods. Some recent developments of HAC include: the FIHC algorithm [44], which generates hierarchical clusters using the frequent itemset concept [3]; information bottleneck-based HAC [116]; an efficient HAC that hybridizes incremental partitional clustering with agglomerative clustering [132]; and a new HAC that uses mutual information as similarity measure [71].

Hierarchical Divisive Clustering

HDC [68] begins with a single cluster containing all objects in a dataset, and then proceeds by recursively splitting up clusters into smaller subclusters. Just like its agglomerative counterpart, the splits in HDC can be based on different ways of measuring similarity between clusters. In its basic form, HDC is less efficient than HAC. This is due to the fact that we have $2^N - 1$ different candidates to be considered for every division into two subsets [136]. For this reason, HDC is less popular than HAC. Hierarchical monothetic clustering algorithms in [76, 79] can be considered as some more recent developments of HDC. In these algorithms, clustering assignment is based on single feature (hence monothetic), as opposed to the more common multiple features.

2.2.2 Partitional Clustering

Unlike hierarchical clustering, partitional clustering generates a single level of clustering. Based on this definition, many existing clustering algorithms can fall under the partitional clustering category. In our case however, we only discuss partitional clustering algorithms that do not obviously belong to other categories (e.g. Fuzzy C-means is discussed in fuzzy clustering and Expectation Maximization is discussed in model-based clustering). Perhaps the most popular partitional clustering algorithm is K-means [85]. The goal of K-means is to partition a dataset into K clusters by minimizing a function of intra-cluster distance. The minimization is performed by applying an iterative procedure of alternating updates of cluster centroids and object-to-cluster assignment until convergence is achieved. In the case of K-means, a cluster centroid is defined as the mean of all objects (in vector form) belonging to the cluster, and the object-to-cluster assignment is crisp or binary in nature.

Besides K-means, partitional clustering can come in many other different forms. Some popular ones are as follows:

- K-medoids [68, 100] is a variant of K-means that replaces the concept of centroid by medoid, which is essentially a set of representative objects in a cluster.
- Leader-based clustering [57] selects several objects in a dataset to be leaders, and then assign the remaining objects into their respective closest leaders to form clusters.
- Nearest neighbor clustering [87] generates clusters by merging objects with their respective nearest neighbors, which are defined based on a threshold similarity value.

Partitional clustering is generally more efficient than its hierarchical counterpart, as

there is only one level of clustering to be considered. However, individually, a partitional clustering algorithm may still suffer from low scalability, sensitivity to noise, and local optima problem. In addition, many partitional clustering algorithms still adopt a crisp object-to-cluster assignment, which may not be very realistic in real-world applications. Some more recent studies on partitional clustering include the works in [65, 119, 141, 30].

2.2.3 Density-based Clustering

Density-based clustering defines dense regions in a dataset as clusters. The goal is then to locate these dense regions in a dataset. Three popular density-based clustering algorithms are DBSCAN [36], OPTICS [4], and DENCLUE [61]. DBSCAN considers objects with dense neighborhood as “core objects”. Based on the idea of core objects, DBSCAN defines the notion of density-connected objects. A cluster is a maximal set of density-connected objects. Clusters can then be found by employing a simple iterative procedure. The outcomes of DBSCAN depend very much on some parameter settings. OPTICS [4] extends DBSCAN by finding dense clusters for different parameter settings simultaneously. The algorithm allows us to see densely populated regions in a dataset in a more general manner than DBSCAN. DENCLUE uses the so-called “influence function” to build a geometric surface capturing regions with high density in a dataset. The algorithm then applies an optimization on this surface to assign objects to dense clusters accordingly. The strong points of density-based clustering are the ability to generate arbitrary-shaped clusters, and the ability to locate noise and outliers (they are generally fall in the region with low density). One main problem in density-based clustering is the difficulty in finding dense regions in high-dimensional data. Some recent developments in density-based clustering include genetic algorithm approach to density-based clustering [82], combination of density-based approach with fuzzy c-means

[8], and density-based clustering for high-dimensional applications [129, 6].

2.2.4 Fuzzy Clustering

Fuzzy clustering applies principles from the fuzzy set theory [142] to generate fuzzy clusters. The fuzzy representation of clusters is flexible enough to allow clusters to overlap one another. This enables fuzzy clustering to capture clustering structure realistically, as most real-world categorizations naturally contain some cluster overlaps (e.g. some mammals lay their eggs). For this reason, fuzzy clustering is often considered to be more informative than its crisp counterpart. There are two main sub-approaches to fuzzy clustering: the fuzzy approach and the possibilistic approach.

Fuzzy Approach

The most well-known fuzzy clustering algorithm based on the fuzzy approach is Fuzzy C-means (FCM) [9]. Similar to K-means, FCM applies an iterative procedure of alternating updates of cluster centroids and object-to-cluster assignment to minimize a function of intra-cluster distance. In FCM however, the object-to-cluster assignment is modeled by fuzzy membership, which conventionally can take any real value between 0 and 1. Because of this, FCM can realistically capture the overlaps among different clusters (i.e. an object does not exclusively belong to one cluster). Most existing fuzzy clustering algorithms in the literature come in the form of improvements to the basic FCM [112, 19, 59, 51]. Despite their informative fuzzy memberships, these FCM-based algorithms are generally vulnerable to outliers due to the common membership constraint imposed in the formulation.

Possibilistic Approach

Another approach to fuzzy clustering is the possibilistic approach. In the literature, it is more commonly known as possibilistic clustering. Possibilistic clustering addresses the vulnerability to outliers of the fuzzy approach. The first and the most popular possibilistic clustering algorithm is Possibilistic C-means (PCM) [72]. PCM relaxes the rigid membership constraint in FCM. This transforms the fuzzy membership in FCM into possibilistic membership, which is more robust to outliers. In addition, possibilistic membership provides a typicality measure of objects in a cluster, which generally can give a more natural representation of clusters than its fuzzy counterparts. One main problem in the possibilistic approach is that it suffers from the notorious coincident-cluster problem [7]. Because of this problem, without a proper initialization, a possibilistic clustering algorithm tends to generate clusters coinciding with one another. Majority of the recent works in possibilistic clustering try to address this issue in order to come up with a stable possibilistic clustering [103, 92, 121, 143].

2.2.5 Graph-based Clustering

In this approach, clustering is performed on the graph representation of a dataset. Generally in the graph representation, objects in a dataset and the connections among them are modeled respectively by vertices and edges of the graph. Clustering is commonly achieved by means of partitioning the graph into several highly connected subgraphs. There are many ways to model the connections among vertices (i.e. objects), and also to generate graph partitions. We will discuss some of the popular ones here. The HCS algorithm [58] first builds the similarity graph of a dataset. This graph is then partitioned by performing a recursive minimum cut procedure to discover subgraphs whose *edge connectivity* exceeds a half of the number of their vertices. In [52], the

frequent itemset concept [3] is used to build hyperedges which connect multiple vertices together. In this algorithm, the average confidence of the frequent *vertex* set is used as the weight of a hyperedge. A hypergraph partitioning technique called HMETIS [66] is then applied to generate clusters. CLICK [114] defines an edge weight as a probabilistic measure capturing if the two connected vertices belong to the same cluster or not. Clusters are then generated through a recursive graph partitioning, followed by refinement procedures of singleton subgraphs adoption and subgraphs merging. In SNN, edges are weighted according to the number of nearest neighbors shared by the two connected vertices [35]. Based on the numbers of connections vertices have, some of them are considered to be representative vertices. Clustering is performed by assigning non-representative vertices into the representative vertices they share the most neighbors to. Graph-based clustering algorithms can be hierarchical or partitional. Some well-known hierarchical graph-based clustering algorithms include Chameleon [67] and AMOEBA [37]. Algorithms like HCS, CLICK, and SNN are examples of partitional graph-based clustering algorithms. Graph-based clustering algorithms are usually good at discovering arbitrary shaped clusters. However, many of them still have drawbacks such as high time complexity and sensitivity to parameter settings. Some recent developments in graph-based clustering include the works in [31], where edge weights are represented by fuzzy sets; and [41], where approximation technique is used to make hierarchical graph-based clustering faster. Another recent effort to improve the efficiency of graph-based clustering can be found in [48].

2.2.6 Model-based Clustering

In model-based clustering, clustering is depicted as an analogous computational process in a relevant mathematical model. In this section, we mainly discuss one im-

portant framework for model-based clustering: mixture density model clustering [93]. In addition, we also briefly go through some other well-known model-based clustering algorithms.

Mixture Density Model Clustering

In mixture density model clustering, every object in a dataset is considered to be generated based on K probability distributions. These K probability distributions are considered to be the clusters in this model. With the type of distribution fixed, e.g. Gaussian, the clustering problem is then seen as the problem of estimating the parameters of the distributions. The optimum solution is the one that follows the maximum likelihood estimation [96]. The Expectation-Maximization (EM) algorithm [93] is one method to get the maximum likelihood solution. EM essentially employs expectation and maximization steps iteratively until convergence is achieved. In the expectation step, the expected cluster memberships of objects are updated. In the maximization step, the distributions' parameters are re-estimated. This iterative process in EM is similar to the one in K-means. Unlike K-means however, EM uses probability distributions to represent clusters. Generally for a mixture density model clustering, a proper selection of probability distributions can be important to determine the clustering outcomes. Some recent developments of the mixture density model include: an accelerated variant of Gaussian mixture model [131], and an improved Gaussian mixture model that automatically decides the number of clusters and reduces the sensitivity to initialization [38, 78].

Other Model-based Clustering

Besides the mixture density model, we have some other popular model-based clustering as follows:

- Self-organizing Map (SOM) [70, 77]: In SOM, clustering is seen as an unsupervised competitive learning to generate a 2-D or 3-D self-organizing feature map.
- ART [16, 17]: Similar to SOM, ART models clustering as an unsupervised competitive learning process. In ART however, we have the process of learning that is similar in fashion as the partitional leader-based clustering (see Section 2.2.2), through the utilization of the resonance concept.
- Clustering based on Genetic Algorithm (GA) [23]: In GA, clustering is modeled as an evolutionary process to select a set of fittest clusters.

2.2.7 Clustering High-dimensional Data

Many real-world data, such as documents and gene expression data, are high dimensional. Without proper care, most of existing clustering algorithms we have discussed so far may not be effective when being applied on high-dimensional data. The reason is that in a high-dimensional space, the object-feature matrix of a dataset becomes very sparse, making it difficult to find any existing cluster pattern. Furthermore, in a high-dimensional space, common distance measures such as the Euclidean distance can no longer capture the notion of dissimilarity anymore [117]. These problems constitute what is commonly referred to as the curses of dimensionality. For this reason, some clustering algorithms have been designed with high-dimensional applications in mind. We discuss three popular approaches for clustering high-dimensional data: clustering with high-dimensional similarity measure, clustering with dimensionality reduction, and co-clustering.

Clustering with High-dimensional Similarity Measure

One relatively straightforward approach to deal with high-dimensional data is to reformulate existing algorithms using similarity measures that are suitable for high-dimensional space. Cosine distance is one such similarity measure [107], and is very popular in the information retrieval field. A few example algorithms that rely on cosine distance for high-dimensional clustering include cosine-based K-means [117], HFCM [95], and UPMGA [32]. One problem with this approach is that when dealing with a large number of features or dimensions, many of these features may actually be noise or information that are irrelevant to the clustering process [88]. Since the computation of a distance measure takes into account all these features, the accuracy and efficiency of this approach may be compromised.

Clustering with Dimensionality Reduction

In this approach, prior to clustering, a dimensionality reduction technique is first applied to remove from consideration dimensions or features that are, in some ways, not too relevant. Clustering is then performed on the reduced-dimensionality space. There are two common methods in this approach:

- Feature selection: In this method, prior to clustering, some salient features are first selected. Clustering is then performed taking these selected features as the dimensions. Feature saliency can be decided based on mutual information [22], chi-square test [11], or inverse document frequency (IDF) [107].
- Feature transformation: In this method, prior to clustering, a dimensionality reduction technique such as principle component analysis (PCA) [140] or singular value decomposition (SVD) [26] is first applied. It is called feature transformation

because the dimensionality reduction technique generates some new features or dimensions that are functions of the original dimensions.

The problem with this approach is the feature selection or transformation takes place somewhat independently from the clustering process itself. In other words, the approach does not provide a mechanism to refine any error that takes place in the feature selection or transformation stage. Because of this, if the feature selection or transformation is inaccurate, the inaccuracy will propagate all the way into the final clustering outcomes without any refinement.

Co-clustering

Co-clustering is defined as simultaneous clustering of objects and features. It puts a strong emphasis on the duality of objects and features, making the clustering process of the two very interdependent. Different fields sometimes use different names for co-clustering: co-clustering in information retrieval, biclustering in bioinformatics, and subspace clustering in data mining. In addition several other names like block clustering and cross associations are often used as well. Figure 2.1 depicts the difference between standard clustering and co-clustering.

Co-clustering is suitable for high-dimensional applications because the simultaneous process of feature clustering can be seen as a dynamic dimensionality reduction of the object clustering space, and vice versa. We call it dynamic because unlike the feature selection or transformation approach, in co-clustering, the dimensionality reduction is continuously refined based on the quality of the object clustering (seeing from the object perspective). This continuous refinement allows co-clustering to provide a more accurate dimensionality reduction than a single feature selection or transformation performed prior to clustering. Another important advantage of co-clustering is that by exploiting

Standard Clustering	Co-clustering
$\begin{bmatrix} 0.9 & 0.8 & 0.1 & \mathbf{c}_1 & 0.1 & 0.05 & 0 \\ 1 & 0.85 & 0.2 & 0 & 0.15 & 0.1 \\ 0.05 & 0 & 0.8 & \mathbf{c}_2 & 0.75 & 0.2 & 0 \\ 0.1 & 0 & 0.95 & 0.8 & 0.1 & 0.05 \\ 0 & 0 & 0 & \mathbf{c}_3 & 0.1 & 1 & 0.9 \\ 0 & 0.1 & 0 & 0 & 1 & 0.8 \end{bmatrix}$	$\begin{bmatrix} 0.9 & \mathbf{cc}_1^{0.8} & 0.1 & \mathbf{cc}_2^{0.1} & 0.05 & \mathbf{cc}_3^0 \\ 1 & 0.85 & 0.2 & 0 & 0.15 & 0.1 \\ 0.05 & \mathbf{cc}_4^0 & 0.8 & \mathbf{cc}_5^{0.75} & 0.2 & \mathbf{cc}_6^0 \\ 0.1 & 0 & 0.95 & 0.8 & 0.1 & 0.05 \\ 0 & \mathbf{cc}_7^0 & 0 & \mathbf{cc}_8^{0.1} & 1 & \mathbf{cc}_9^{0.9} \\ 0 & 0.1 & 0 & 0 & 1 & 0.8 \end{bmatrix}$

Figure 2.1: Standard Clustering vs. Co-clustering

the duality of object and features, the resulting feature clusters should closely capture the contents of the object clusters [29], and vice versa. For example in the case of document-word co-clustering, the word clusters are commonly utilized to generate the corresponding document clusters' descriptions. This makes co-clustering capable of generating interpretable clusters.

There are many different approaches to achieve co-clustering. The following are some of the most prominent ones:

- Hierarchical co-clustering: Similar to its hierarchical clustering counterpart, hierarchical co-clustering generates different levels co-clustering arranged in hierarchical structure. The algorithms in [56, 122] are example of hierarchical divisive co-clustering, while the algorithm in [89] is an example of hierarchical agglomerative co-clustering.
- Graph-based co-clustering: In this approach, the interrelation between objects and features is represented by a bipartite graph. Co-clusters are then generated by applying a graph partitioning technique on the bipartite graph. Algorithms that adopt this approach can be found in [27, 46].

- Probabilistic co-clustering: In this case, co-clusters are modeled by probability distributions of objects and features. One popular algorithm falls under this category is the Information-theoretic Co-clustering [29], in which co-clustering is seen as a problem of finding a compressed object-feature joint probability distribution that minimizes mutual information loss. Other examples of probabilistic co-clustering include [5, 113].
- Fuzzy co-clustering: This approach extends the concepts of fuzzy clustering into a co-clustering domain. It preserves the informative nature of fuzzy clustering through the representation of co-clusters by fuzzy object and feature memberships. Some fuzzy co-clustering algorithms include Fuzzy Clustering of Categorical and Multivariate data (FCCM) [101], Fuzzy Co-clustering of Documents and Keywords (Fuzzy CoDoK) [75], and Fuzzy Simultaneous KeyWord Identification and Clustering of text documents (FSKWIC) [42].
- Co-clustering based on matrix factorization: The goal of this approach is to factorize an object-feature matrix into several cluster assignment matrices. Co-clustering corresponds to the factorization that minimizes any information loss. Prominent matrix factorization-based co-clustering algorithms include NMF [138, 30], co-clustering based on block value decomposition [86], ASI [81], and document clustering based on concept factorization [137].
- Co-clustering through subspace search: This approach generate object clusters on different subspaces of dataset. The goal is then to find the best clusters and the corresponding subspaces (which are equivalent to the corresponding feature clusters). Certain subspace evaluation measures, which are generally based on the quality of the corresponding clusters, are used to guide the subspace search process. Some popular algorithms under this category are CLIQUE [2], PROCLUS

[1], and MAFIA [47].

- Other co-clustering approaches: There are many other co-clustering approaches which cannot be categorized into any of the above. Examples include the works in [20, 21], which use the concept of minimum sum-squared residue to generate co-clusters, sleeved co-clustering in [94], which defines the concept of sleeve-width to find the optimum co-clusters, and the spectral approach to co-clustering [69]. [88, 120] provide comprehensive surveys of co-clustering.

2.2.8 Why Fuzzy-possibilistic Co-clustering

In this section, we explain why we need a new fuzzy-possibilistic co-clustering framework despite so many existing clustering algorithms available. We have explained how co-clustering, compared to other approaches, provides arguably the most solid framework for high-dimensional applications. The exploitation of the duality of objects and features makes co-clustering effective against the curse of dimensionality, as well as capable to automatically generate some descriptions of the resulting clusters. These are inline with our objectives of effectiveness in high dimension and interpretability of clusters. To capture realistic categorizations of real-world data, which generally contain overlaps, it is essential to have soft, rather than crisp, co-clustering. There are several options for soft approach to co-clustering, such as fuzzy, possibilistic, probabilistic, and some matrix-based approaches. However, of all these, to the best of our knowledge, only the possibilistic approach has been popularly acclaimed for the combined robustness to outliers and intuitive typicality representation of cluster membership [72, 143, 121]. Hence, by basing on the possibilistic approach, we can have a soft co-clustering for interpretable high-dimensional data analysis that is robust to outliers, and also able to capture natural representations of co-clusters. Relying on the possibilistic approach alone however,

may lead to instability caused by the coincident cluster problem as discussed in Section 2.2.4. Incorporating fuzzy memberships into a possibilistic formulation has been shown to be able to provide the necessary stability for possibilistic clustering [103, 92, 143]. Furthermore, since the fuzzy approach is very related to the possibilistic approach, integration of the two can be done in a more straightforward manner than integration with other more distant approaches. For this reason, we propose fuzzy-possibilistic co-clustering to have a stable soft co-clustering framework that simultaneously addresses the aspects of applicability in high-dimensional space, quality of cluster representations, robustness to outliers, and interpretability of clusters.

2.3 Relevant Works

Several existing models from the fuzzy, possibilistic, and co-clustering approaches of clustering have provided the needed foundations for our proposed fuzzy-possibilistic co-clustering framework. In this section, we provide the detail discussions on these models, as they can be helpful to better understand our discussions in the subsequent chapters. Specifically, we have four important models. Since our soft co-clustering is based on the fuzzy and possibilistic approaches, the first and second models are naturally Fuzzy C-means (FCM) and Possibilistic C-means (PCM). The third model is Possibilistic-fuzzy C-means (PFCM) [103]. PFCM hybridizes the formulations of FCM and PCM to address simultaneously the vulnerability to outliers in the former, and the coincident-cluster problem in the latter. Although PFCM is not the only existing hybrid fuzzy-possibilistic clustering algorithm [92, 143], it arguably provides the most natural merge between the two clustering elements. This in turns makes it more flexible for further expansion, in our case, to co-clustering. The fourth is a fuzzy co-clustering model called Fuzzy Clustering for Categorical and Multivariate data (FCCM) [101].

Compared to other well-known existing fuzzy co-clustering models [43, 42], FCCM arguably gives the most solid theoretical foundation where all the necessary elements can be mathematically derived.

We use some mathematical notations when discussing these models. Table 2.1 explains all the mathematical notations used.

Table 2.1: List of Mathematical Notations

Notation	Description
C, N, K	Numbers of (co-)clusters, objects, and features
u_{ci}, v_{cj}	Fuzzy object and feature memberships
x_{ci}	Possibilistic object memberships
o_i	Objects in a dataset
p_c	Cluster centroid or prototype
$Dist()$	Distance function
d_{ij}	Relatedness measure between an object and a feature ($d_{ij} \geq 0$)
$T_u, m, \gamma_c, \eta, T_v$	Clustering and co-clustering user-defined parameters
τ	Number of iterations
τ_{max}	Maximum number of iterations parameter
ϵ	Convergence indicator parameter

2.3.1 The FCM Model

The goal of FCM is to partition objects into a set of clusters based on their distances to the clusters. At the same time FCM also tries to capture the overlaps among the different clusters by representing the object-to-cluster assignments by fuzzy memberships. The higher the fuzzy membership, the more the object belongs to the cluster. One version of the FCM algorithm [97] translates the clustering problem into a problem of minimizing the objective function in Eq. (2.1) subject to the membership constraint in Eq. (2.2). Note that here we are using the version of FCM in [97], which may be considered as a variant of the general *maximum-entropy clustering* framework studied in [108, 64]. This version of FCM is more closely-related to our proposed algorithms

and is actually fundamentally equivalent to the original FCM in [9].

$$J_{FCM} = \sum_{c=1}^C \sum_{i=1}^N u_{ci} \text{Dist}(o_i, p_c) + T_u \sum_{c=1}^C \sum_{i=1}^N u_{ci} \ln u_{ci} \quad (2.1)$$

$$\sum_{c=1}^C u_{ci} = 1 \text{ for } i = 1, 2, \dots, N \quad (2.2)$$

The term $\text{Dist}(x_i, p_c)$ denotes any suitable distance function for the given problem. Minimizing the first term in Eq. (2.1) forces each object to have a higher membership to a cluster it has a closer distance to. In this way, similar objects will be clustered together. The second term of Eq. (2.1) is the fuzzy entropy [97], which controls the fuzziness of the resulting clusters. In this case, one can increase the fuzziness of the resulting clusters by setting a higher parameter T_u value. If $\text{Dist}(o_i, p_c)$ is the Euclidean distance, the minimization of Eq. (2.1) can be solved by alternatively updating Eqs. (2.3) and (2.4) until the convergence is achieved.

$$u_{ci} = \frac{\exp \left\{ -\frac{\text{Dist}(o_i, p_c)}{T_u} \right\}}{\sum_{f=1}^C \exp \left\{ -\frac{\text{Dist}(o_i, p_f)}{T_u} \right\}} \quad (2.3)$$

$$p_c = \frac{\sum_{i=1}^N u_{ci} o_i}{\sum_{i=1}^N u_{ci}} \quad (2.4)$$

Based on Eq. (2.3), an object membership u_{ci} in FCM essentially measures the similarity between an object i and a cluster c , relative to the similarities between the object i and all the other clusters. This notion of “relative to other clusters” is uniquely associated with the membership constraint in Eq. (2.2), which is also known in the literature as the Ruspini’s condition due to the work by Ruspini in [110]. This constraint, from its definition, forces the determination of a membership to a cluster to be depen-

dent on the states of the other clusters. This membership characteristic is important in understanding the difference between fuzzy clustering and possibilistic clustering. To avoid confusion in later discussions, from this point onward, we will refer to a clustering membership function with such a characteristic as the *partitioning membership*.

2.3.2 The PCM Model

PCM was developed to address the vulnerability of FCM to outliers. To demonstrate this vulnerability, let us consider a clustering problem depicted in Fig. 2.2. In Fig. 2.2, P can be considered an outlier because its position is significantly remote from all the other objects or points. Q can be considered as a non-outlier but similar to P , it has an equal distance to the (prototypes of) clusters $C1$ and $C2$. Despite both P and Q being right in the middle of the two clusters, intuitively it would be more appropriate if Q has greater membership values than P to $C1$ and $C2$. This is obvious because P is physically farther away from the two clusters than Q is. If we apply FCM to this problem however, we will end up with both P and Q having 0.5 membership to each $C1$ and $C2$ (i.e $u_{1P} = u_{1Q} = 0.5$ and $u_{2P} = u_{2Q} = 0.5$). The root of the problem lies in the constraint in Eq. (2.2), or equivalently the *partitioning membership* of FCM. Let us revisit Eq. (2.3) and express it in the form of $u_{ci} = \frac{A_{ci}}{\sum_{f=1}^C A_{fi}}$, where $A_{ci} = \exp \left\{ -\frac{Dist(x_i, p_c)}{T_u} \right\}$. The distance from P to $C1$ is greater than the distance from Q to $C1$, resulting in the similarity $A_{1P} < A_{1Q}$. But this similarity between P and $C1$, relative to the similarities between P and all the other clusters (as captured by $\frac{A_{1P}}{\sum_{f=1}^C A_{fP}}$) is equal to the similarity between Q and $C1$ (A_{1Q}) relative to the similarities between Q and all the other clusters ($\frac{A_{2P}}{\sum_{f=1}^C A_{fQ}}$). For this reason, both P and Q have exactly the same memberships. It should be evident from this discussion that the cause of this misinterpretation is the notion of "relative to other clusters" in the partitioning

membership, which, as we have mentioned, is the consequence of complying with the membership constraint in Eq. (2.2).

Figure 2.2: Outlier Problem

PCM addresses this limitation in FCM by removing the constraint from the formulation. The goal of PCM is to minimize the objective function in Eq. (2.5), subject to no membership constraint whatsoever. Here, we use the variant of PCM proposed in [73] because it is more closely-related to our proposed algorithms.

$$J_{PCM} = \sum_{c=1}^C \sum_{i=1}^N x_{ci} Dist(o_i, p_c) + \sum_{c=1}^C \left\{ \gamma_c \sum_{i=1}^N (x_{ci} \ln x_{ci} - x_{ci}) \right\} \quad (2.5)$$

The first term of Eq. (2.5) is exactly the same as the first term of Eq. (2.1). Without imposing any constraint on the x_{ci} membership, the minimization of this term forces the membership values to be as small as possible. The minimization of the second term of Eq. (2.5) on the other hand, balances this by pushing the membership values to be as large as possible. The parameter γ_c represents the distance threshold that is used to decide if any particular object still belongs to the cluster c or not. Therefore, a large value of γ_c will make x_{ci} for all $i = 1, \dots, N$ to take on their maximum values.

If $Dist(o_i, p_c)$ is a Euclidean distance, Eq. (2.5) can be minimized by alternatively updating Eq. (2.6) below and Eq. (2.4) shown in Section 2.3.1.

$$x_{ci} = \exp \left\{ -\frac{Dist(o_i, p_c)}{\gamma_c} \right\} \quad (2.6)$$

Based on Eq. (2.6), it is obvious that the value of a possibilistic membership x_{ci} depends only on the distance between an object i and an cluster c . Such a measure is called the typicality measure of object i in cluster c . From this, we can see that, unlike the partitioning membership, the possibilistic membership is independent from the states of other clusters. Consequently, an outlier like the object P in Fig. 2.2 would be assigned very low possibilistic membership values to $C1$ and $C2$ because its distances to both clusters are large. This explains how a possibilistic clustering, due to the characteristic of its possibilistic membership, is able to minimize the impact of outliers present in a dataset [72]. At the same time, being a typicality measure, the possibilistic membership x_{ci} offers a better interpretability of cluster membership than the fuzzy counterpart. We explain this in more details in Section 4.2 in Chapter 4.

Despite their known robustness, many possibilistic clustering algorithms including PCM suffer one notorious problem generally referred to as the *coincident cluster* problem [7]. To understand this problem, let us decompose J_{PCM} in Eq. (2.5) into J_{PCM_1} , J_{PCM_2} , $\dots$, J_{PCM_C} . Since the latter are independent from one another (due to the absence of the constraint in Eq. (2.2)), the global minimum of J_{PCM} can be achieved if $J_{PCM_1} = J_{PCM_2} = \dots = J_{PCM_C} = \min_{c \in 1, \dots, C} J_{PCM_c}$ [121]. This corresponds to a condition where all the clusters coinciding exactly in the same location. Because of this problem, without any proper initialization, PCM, instead of discovering the intended clusters, often results in coincident clusters. This coincident cluster problem does not affect FCM due to the influence of the constraint in Eq. (2.2).

2.3.3 The PFCM Model

The goal of PFCM is to minimize the objective function in Eq. (2.7), subject to the constraint in Eq. (2.2). Note that the style of PFCM follows the original versions of FCM and PCM in [9] and [72] respectively (i.e. not the versions of FCM and PCM that we discussed in the two previous sections). In this case, the parameter $m \geq 1$ acts as the fuzzifier. Given the constraint, a large value of m will cause $u_{ci} \approx \frac{1}{C}$ for all $c = 1, \dots, C$ and $i = 1, \dots, N$, creating a total fuzziness in the resulting clusters. When $m = 1$, the membership u_{ci} will take on a crisp value (i.e. 0 or 1). The parameter $\eta \geq 1$ serves a similar but not exactly the same purpose as the parameter m in FCM [73]. Setting η to be large will cause a total fuzziness in all the memberships, just like in the case of a large value of m . If $\eta = 1$, x_{ci} will take on a crisp value, i.e. 1 if $\text{Dist}(o_i, p_c) < \gamma_c$, and 0 if $\text{Dist}(o_i, p_c) > \gamma_c$. The parameters T_F and T_P are used to adjust the balance between the influences of FCM and PCM. A large T_F will make FCM more dominant, while a large T_P will make PCM more dominant. The parameters m and η can also be used for the same purpose. A small m will make FCM more dominant, while a small η will make PCM more dominant. By applying the Lagrange multiplier method, we can get the update rules for u_{ci} , x_{ci} , and p_c in Eqs. 2.8-2.10. With the right setting of parameters, PFCM can simultaneously exhibit the PCM's robustness to outliers and FCM's invulnerability to the coincident-cluster problem [103].

$$J_{PFCM} = \sum_{c=1}^C \sum_{i=1}^N \{T_F u_{ci}^m + T_P x_{ci}^\eta\} \text{Dist}(o_i, p_c) + \sum_{c=1}^C \gamma_c \sum_{i=1}^N (1 - x_{ci})^\eta \quad (2.7)$$

$$u_{ci} = \left[\sum_{e=1}^C \left\{ \frac{\text{Dist}(o_i, p_c)}{\text{Dist}(o_i, p_e)} \right\}^{\frac{1}{m-1}} \right]^{-1} \quad (2.8)$$

$$x_{ci} = \frac{1}{1 + \left\{ \frac{T_P \text{Dist}(o_i, p_c)}{\gamma_c} \right\}^{\frac{1}{\eta-1}}} \quad (2.9)$$

$$p_c = \frac{\sum_{i=1}^N u_{ci}^m o_i}{\sum_{i=1}^N u_{ci}^m} \quad (2.10)$$

From PFCM's distance-based objective function in Eq. (2.7), we can see that the algorithm, just like its predecessors FCM and PCM, may not be suitable for high-dimensional applications. This inspires us to bring this model to the co-clustering domain. As we discussed in Section 2.2, co-clustering is known to be effective for categorization of high-dimensional data.

2.3.4 The FCCM Model

There are two algorithms that constitute the FCCM model: FCCM and its variant Fuzzy CoDoK [75]. Fuzzy CoDoK is formulated with the purpose of categorization of large datasets. We discuss them both here.

The goal of the FCCM algorithm is to maximize the objective function in Eq. (2.11) under the object membership constraint in Eq. (2.2) in Section 2.3.1 and the feature membership constraint in Eq. (2.12) below.

$$J_{FCCM} = \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj} d_{ij} - T_u \sum_{c=1}^C \sum_{i=1}^N u_{ci} \ln u_{ci} - T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj} \ln v_{cj} \quad (2.11)$$

$$\sum_{j=1}^K v_{cj} = 1 \text{ for } c = 1, \dots, C \quad (2.12)$$

Notice that we have the separate object and feature memberships u_{ci} and v_{cj} respec-

tively, a clear indication that the algorithm simultaneously clusters both objects and features. The first term in Eq. (2.11) is called the *degree of aggregation* [101]. The maximization of this term makes highly related object and feature, as indicated by a high value of d_{ij} , to be co-clustered together. This is because the maximization would force the value of v_{cj} to be high whenever the values of d_{ij} and u_{ci} are simultaneously high, or alternatively the value of u_{ci} to be high whenever the values of d_{ij} and v_{cj} are simultaneously high. By using the degree of aggregation, instead of relying on the concept of centroid, FCCM is not constrained to find only spherical-shaped clusters. The second and third terms in Eq. (2.11) are the entropy terms, which serve as fuzzifiers of object and feature memberships. These terms are used to control the levels of fuzziness of the memberships. From these entropy terms in its objective function, we can consider FCCM as a fuzzy co-clustering extension of the maximum entropy clustering framework in [108, 64]. The constraint in Eq. (2.12) has a different orientation from the constraint in Eq. (2.2). This results in a different type of membership as will be discussed shortly.

There are two important user-defined parameters in FCCM: T_u and T_v . They are used to adjust the levels of fuzziness in the object membership u_{ci} and the feature membership v_{cj} respectively. A very high value of T_u results in a very fuzzy u_{ci} . Considering the constraint in Eq. (2.2), this is equivalent to having $u_{ci} \approx \frac{1}{C}$ for all $c = 1, \dots, C$ and $i = 1, \dots, N$. A very small value of T_u results in a very crisp u_{ci} , i.e. its value is close to either 0 or 1. Adjusting T_v gives these similar affects to v_{cj} .

The FCCM objective function J_{FCCM} in Eq. (2.11) can be maximized by alternatively updating the following membership equations until convergence is achieved:

$$u_{ci} = \frac{\exp \left\{ \frac{\sum_{j=1}^K v_{cj} d_{ij}}{T_u} \right\}}{\sum_{f=1}^C \exp \left\{ \frac{\sum_{j=1}^K v_{fj} d_{ij}}{T_u} \right\}} \quad (2.13)$$

$$v_{cj} = \frac{\exp \left\{ \frac{\sum_{i=1}^N u_{ci} d_{ij}}{T_v} \right\}}{\sum_{q=1}^K \exp \left\{ \frac{\sum_{i=1}^N u_{ci} d_{iq}}{T_v} \right\}} \quad (2.14)$$

By observing the structures of the two update membership equations in Eqs. (2.13) and (2.14), we can see that the interactions among u_{ci} , v_{cj} , and d_{ij} in J_{FCCM} allows FCCM to tackle the curse of dimensionality and sparseness problems in high-dimensional data. When computing object memberships u_{ci} by Eq. (2.13), the feature memberships terms v_{cj} serve to limit the scope of analysis to the most prominent features only, as opposed to all the features (and vice versa when computing the v_{cj} by Eq. (2.14)). This scheme locally reduces the dimensionality and hides the sparseness of data.

By conforming to the membership constraint in Eq. (2.2), the object membership u_{ci} of FCCM becomes partitioning in nature. This is also indicated by the resemblance between Eqs. (2.13) and (2.3). On the other hand, the FCCM's feature membership in Eq. (2.14) is of different nature from any membership we have discussed so far. One can observe from the equation that v_{cj} is a measure of similarity between a feature j and a co-cluster c , relative to the similarities between all the other features and the co-cluster c . To comprehend this, we first express Eq. (2.14) in the form of $v_{cj} = \frac{A_{cj}}{\sum_{q=1}^K A_{cq}}$, where $A_{cj} = \exp \left\{ \frac{\sum_{i=1}^N u_{ci} d_{ij}}{T_v} \right\}$. Since A_{cj} captures the similarity between the feature j and the co-cluster c , we have $\frac{A_{cj}}{\sum_{q=1}^K A_{cq}}$ captures the similarity between the feature j and the co-cluster c , relative to the similarities between all the other features and the co-cluster c . Like in the case of the partitioning membership, the notion of “relative to other features” is uniquely associated with the constraint in Eq. (2.12), which forces the determination of a feature membership to one co-cluster to be dependent on the states of the other features in the same co-cluster. From this point onward, we will refer to a membership of this kind as the *ranking membership*, because such a membership reflects the rank (in

this case, the feature's rank) of importance in a given co-cluster. Unlike the partitioning membership but like the possibilistic membership, the ranking membership does not depend on the states of other clusters. Different from the possibilistic membership however, the ranking membership does depend on the states of other features in the given cluster (or co-cluster).

It is argued in [75] that FCCM is not suitable for large datasets. The reason can be observed from the FCCM's update membership equation in Eq. (2.14). In this equation, both the numerator and the denominator are in the form of $\frac{\sum_{i=1}^N u_{ci} d_{ij}}{T_v}$. When dealing with large datasets, i.e. with a large value of N , this exponential function may be equal to a very large number. Such a very large number can cause a computational overflow once its value passes the maximum limit allowable by a computer [75]. Because of this problem, it is prohibitive to apply FCCM for datasets containing more than a few hundreds objects. This is the motivation behind Fuzzy CoDoK. In Fuzzy CoDoK, the objective function is modified into Eq. (2.15). The goal is to maximize this function, subject to the constraints in Eqs. (2.2) and (2.12). By applying the Lagrange multiplier method, we can derive the update equations for Fuzzy CoDoK in Eqs. (2.16) and (2.17). It can be seen that these two equations do not involve any exponential function. For this reason, Fuzzy CoDoK does not suffer the computational overflow problem like FCCM.

$$J_{FCODOK} = \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj} d_{ij} - T_u \sum_{c=1}^C \sum_{i=1}^N u_{ci}^2 - T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj}^2 \quad (2.15)$$

$$u_{ci} = \frac{1}{C} + \frac{1}{T_u} \left(\sum_{j=1}^K v_{cj} d_{ij} - \frac{1}{C} \sum_{f=1}^C \sum_{j=1}^K v_{fj} d_{ij} \right) \quad (2.16)$$

$$v_{cj} = \frac{1}{K} + \frac{1}{T_v} \left(\sum_{i=1}^N u_{ci} d_{ij} - \frac{1}{K} \sum_{q=1}^K \sum_{i=1}^N u_{ci} d_{iq} \right) \quad (2.17)$$

As fuzzy co-clustering algorithms, both FCCM and Fuzzy CoDoK can handle high-dimensional datasets quite effectively [75]. However, incorporating possibilistic memberships into their formulations will improve their robustness to outliers and the interpretability of the resulting co-clusters. We shall see in Chapter 3 that the FCCM model, based on which FCCM and Fuzzy CoDoK are formulated, does not provide a firm theoretical foundation for extension to fuzzy-possibilistic co-clustering. In Chapter 3, we show how we modify the existing FCCM model, to provide a foundation for our proposed fuzzy-possibilistic co-clustering framework.

2.4 Conclusions

We have presented a literature review of a number of popular data clustering approaches. Based on this, we have justified how our work can further contribute to the progress of clustering study in general; in the fuzzy-possibilistic co-clustering framework, we can have soft clustering algorithms that simultaneously are suitable for high-dimensional applications (due to co-clustering), robust to outliers (due to possibilistic membership), rich in representations (due to the combined fuzzy-possibilistic memberships), and able to produce highly-interpretable results (due to the feature clusters generated by co-clustering). In addition, we have also given some detailed discussions on some relevant existing models that lay the foundation for our work.

As a closing remark for this chapter, we are aware of recent interests in possibilistic-based co-clustering algorithms in bioinformatics, such as the works in [39] in late 2006 and [14] in early 2007. The method used in [39] is different from our proposed algorithms

in a number of aspects, including: the way co-clusters are generated, i.e. one-by-one in their case vs. simultaneously in our case; and the utilization of fuzzy memberships in our case, which is absent in their case. We have only managed to access the abstract of [14], and hence we are unable to make any comment on it for now. In any case, considering the fact that these two related algorithms were developed almost simultaneously with ours, we will not include them in our current study.

Chapter 3

The Dual-partitioning Approach to Fuzzy Co-clustering

Building the Foundation for Fuzzy-Possibilistic Co-clustering

3.1 Overview

In this chapter, we report the details on the development of the dual-partitioning approach to fuzzy co-clustering. In this approach, fuzzy co-clustering is achieved by simultaneously performing Fuzzy C-means-style (FCM-style) [9] clustering on objects and features. The main motivation behind the development of the approach is to build a theoretical foundation for fuzzy-possibilistic co-clustering, which is discussed in Chapters 4 and 5. We will explain later that the existing prominent fuzzy co-clustering model, i.e. the FCCM model, which serves as the basis for algorithms like FCCM and Fuzzy CoDoK [101, 75] is insufficient for this purpose. Additionally, the dual-partitioning approach can offer some benefits over the existing FCCM model, such as a more natural

representation of feature clusters in certain situations.

We first provide the formal definition of the dual-partitioning approach in Section 3.2, and attempt to bring this definition into the context of the existing FCCM model in Section 3.3. From this attempt, a limitation of this existing formulation that prevents possibly useful expansions becomes obvious. In Section 3.4, we propose three possible methods to address this limitation. For each of these methods, we formulate the corresponding models for fuzzy co-clustering based on the dual-partitioning approach. Section 3.5 discusses the advantages and disadvantages of the dual-partitioning approach over the existing FCCM model. We show and discuss some experimental results on benchmark document datasets in Section 3.6. Section 3.7 concludes the chapter.

3.2 The Definition of the Dual-partitioning Approach

Table 3.1 provides the list of all mathematical notations used in this chapter. We define the dual-partitioning approach to fuzzy co-clustering (or in short, the dual-partitioning approach) as follows:

Definition 2 *The dual-partitioning approach to fuzzy co-clustering is a process of simultaneous fuzzy clustering of objects and features, with object and feature memberships observing the constraints in Eq. (2.2) in Section 2.3.1 of Chapter 2 and Eq. (3.1) below respectively.*

$$\sum_{c=1}^C v_{cj} = 1, \text{ for } j = 1, \dots, K \quad (3.1)$$

The membership constraints in Eqs. (2.2) and (3.1) are of the same type as the membership constraint in Fuzzy C-means (FCM) [9] (see Section 2.3.1 of Chapter 2). Recall from Chapter 2 that this constraint is often referred to as the Ruspini's condition [110],

Table 3.1: List of Mathematical Notations

Notation	Description
C, N, K	Numbers of (co-)clusters, objects, and features
u_{ci}, v_{cj}	Fuzzy object and feature memberships
d_{ij}	Relatedness measure between an object and a feature ($d_{ij} \geq 0$)
$\mathbf{D}$	A $N \times K$ d_{ij} matrix
T_u, T_v	Co-clustering user-defined parameters
τ	Number of iterations
τ_{max}	Maximum number of iterations parameter
ϵ	Convergence indicator parameter

and fuzzy membership that observes such a condition captures the partitioning structure of data in a dataset [33], i.e. the partitioning membership. For this reason, we use the name the dual-partitioning approach. At this point, we would like to highlight the difference between the dual-partitioning approach and the existing FCCM model. In the FCCM model, the feature membership's constraint follows Eq. (2.12) in Section 2.3.4 of Chapter 2. Again we recall from Chapter 2 that fuzzy membership that observes this constraint captures the ranking structure of data in a dataset, i.e. the ranking membership. We will discuss in more details the consequences of the difference between the dual-partitioning approach and the existing FCCM model later in Section 3.5.

3.3 The Limitation of the FCCM Model

Based on Definition 2, one way to achieve the dual-partitioning approach in the context of the existing FCCM model is by optimizing the objective function J_{FCCM} in Eq. (2.11), subject to the constraints in Eqs. (2.2) and (3.1). Let us refer to such a dual-partitioning model as the DP-FCCM model, for Dual-Partitioning FCCM. If we apply the Lagrange multiplier method on the DP-FCCM model, we get the following update

membership equations:

$$u_{ci} = \frac{\exp \left\{ \frac{\sum_{j=1}^K v_{cj} d_{ij}}{T_u} \right\}}{\sum_{f=1}^C \exp \left\{ \frac{\sum_{j=1}^K v_{fj} d_{ij}}{T_u} \right\}} \quad (3.2)$$

$$v_{cj} = \frac{\exp \left\{ \frac{\sum_{i=1}^N u_{ci} d_{ij}}{T_v} \right\}}{\sum_{f=1}^C \exp \left\{ \frac{\sum_{i=1}^N u_{fi} d_{ij}}{T_v} \right\}} \quad (3.3)$$

Note that Eq. (3.2) is exactly the same as Eq. (2.13) in Chapter 2. It is repeated here for the sake of readability. In [101, 127], it is reported that the DP-FCCM model is not functional, because its optimization tends to generate a single co-cluster that contains all objects and features. Mathematically, this single co-cluster outcome corresponds to the object and feature memberships configuration below, where $\mathbf{U}$ denotes a $C \times N$ matrix with rows correspond to clusters and columns correspond to objects, and $\mathbf{V}$ denotes a $C \times K$ matrix with rows correspond to clusters and columns correspond to features.

$$\mathbf{U} = \begin{bmatrix} 1 & \dots & 1 \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{V} = \begin{bmatrix} 1 & \dots & 1 \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}$$

To understand the cause of this problem, let us represent the degree of aggregation term (i.e. the first term) of J_{FCCM} in Eq. (2.11) as follows:

$$\mathbf{M} : \mathbf{D} = \begin{bmatrix} m_{11} & \cdots & m_{1K} \\ \vdots & \ddots & \vdots \\ m_{N1} & \cdots & m_{NK} \end{bmatrix} : \begin{bmatrix} d_{11} & \cdots & d_{1K} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{bmatrix}$$

where $:$ denotes the component-wise inner product of two matrices, i.e. $\mathbf{A} : \mathbf{B} = \sum_x \sum_y a_{xy} b_{xy}$; and $m_{ij} = \sum_{c=1}^C u_{ci} v_{cj}$. Based on this representation, it can be seen that, regardless of $\mathbf{D}$ (i.e. the dataset), the degree of aggregation is maximum if every element of the matrix $\mathbf{M}$ is equal to its maximum possible value; note that throughout this thesis, we assume that $d_{ij} \geq 0$, as is the case in many real-world categorization problems such as document-word co-clustering [63]. By considering the DP-FCCM membership equations in Eqs. (3.2) and (3.3), it can be shown that such a maximum value in DP-FCCM is equal to 1, i.e. $m_{ij} = 1$. In other words, in the DP-FCCM model, the maximization of the degree of aggregation, regardless of the dataset, strives only to make the matrix $\mathbf{M}$ above equal to a unit matrix. It is not difficult to show, taking into consideration the membership constraints in Eqs. (2.2) and (3.1), that $\mathbf{M}$ is a unit matrix iff the object and feature memberships are equal to the $\mathbf{U}$ and $\mathbf{V}$ shown above, which correspond to the single co-cluster outcome. For this reason, a proper fuzzy co-clustering cannot be achieved using the DP-FCCM model. During the optimization, the fuzzifier components of J_{FCCM} are also maximized. However, these fuzzifiers are dataset-independent, and therefore, their sole purpose is to act as a mean to adjust the fuzziness levels of the memberships. Hence, to get correct co-clusters, it is essential that the maximization of the degree of aggregation points to the right direction.

We now discuss why, despite the flaw in the J_{FCCM} as discussed above, FCCM (or Fuzzy CoDoK) can still work at all. Recall that the feature membership of this algorithm observes the membership constraint in Eq. (2.12). Taking into consideration

the membership constraints in Eqs. (2.2) and (2.12), the following equality is always valid in FCCM:

$$\sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{cj} v_{cj} = N$$

This is the same as saying that the summation of all elements of the $\mathbf{M}$ matrix of FCCM is always equal to a constant (i.e. $\sum_{i=1}^N \sum_{j=1}^K m_{ij} = \text{constant}$), regardless of the individual membership values. If we revisit the DP-FCCM model, we find that such an equality is not valid there, that is, $\sum_{i=1}^N \sum_{j=1}^K m_{ij} \neq \text{constant}$. In fact, in this model, the maximization of the degree of aggregation is biased towards membership assignments with larger $\sum_{i=1}^N \sum_{j=1}^K m_{ij}$ values, because, as explained above, this maximization tries to increase the value of every element m_{ij} as much as possible. This, as we have seen, causes the optimization to converge to a single co-cluster, at which we have the maximum possible value of $\sum_{i=1}^N \sum_{j=1}^K m_{ij}$ (i.e. when $m_{ij} = 1 \mid \forall i, j$ in the case of DP-FCCM) [127]. This single co-cluster problem does not happen in FCCM because their $\sum_{i=1}^N \sum_{j=1}^K m_{ij}$ is maintained at a constant value.

In FCCM, the property $\sum_{i=1}^N \sum_{j=1}^K m_{ij} = \text{constant}$ is ensured by their combination of membership constraints, i.e. Eqs. (2.2) and (3.1). For any general combination of membership constraints however, there is no guarantee that such a property can be maintained, just like in the dual-partitioning DP-FCCM model above. Therefore, it is not possible to directly expand the existing FCCM model, by incorporating different types of memberships into the objective function.

3.4 The Proposed Models for the Dual-partitioning Approach

Continuing from our discussion in the previous section, the challenge therefore is to develop a new model, based on the existing FCCM model, in which, for any given membership constraints, there is no unwanted bias in the maximization of the degree of aggregation term of the objective function. In this section, we propose three possible methods to achieve this. For each method, we propose the corresponding model and algorithm for fuzzy co-clustering based on the dual-partitioning approach. Note that, it is not necessary for these models to be based on the dual-partitioning approach, any approach as long as the bias is absent should be functional. In this chapter, we focus our effort in making the dual-partitioning approach work because such an approach can be considered as a fundamental theoretical foundation for any further expansion, considering the approach is essentially equivalent to a dual FCM, which itself (i.e. FCM) is well-regarded as the most fundamental fuzzy clustering model.

3.4.1 The First Model: The FCR Algorithm

In this section, we discuss Fuzzy Co-clustering with the Ruspini's condition (FCR), the first fuzzy co-clustering algorithm we propose based on the dual-partitioning approach [127]. We first go through the structure of FCR, and then continue by explaining the concept behind the algorithm. Afterwards, we provide the proof of convergence of FCR and a simple illustration to demonstrate the feasibility of the algorithm.

The Algorithm

Consider the following objective function:

$$\begin{aligned}
 J_{FCR} = & \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj} d_{ij} - \frac{\sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj}}{NK} \sum_{i=1}^N \sum_{j=1}^K d_{ij} - \\
 & T_u \sum_{c=1}^C \sum_{i=1}^N u_{ci} \ln u_{ci} - T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj} \ln v_{cj}
 \end{aligned} \tag{3.4}$$

The difference between J_{FCR} and J_{FCCM} in Eq. (2.11) lies in the additional second term in the former. We will now explain how, by incorporating this second term, we can emulate the $\sum_{i=1}^N \sum_{j=1}^K m_{ij} = \text{constant}$ property of the existing FCCM model, and thus remove the unwanted bias when maximizing the degree of aggregation as discussed in Section 3.3. Let us represent the first and the second terms of J_{FCR} as follows:

$$\begin{aligned}
 (\mathbf{M} : \mathbf{D}) + BD &= \left(\begin{bmatrix} m_{11} & \cdots & m_{1K} \\ \vdots & \ddots & \vdots \\ m_{N1} & \cdots & m_{NK} \end{bmatrix} : \begin{bmatrix} d_{11} & \cdots & d_{1K} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{bmatrix} \right) + \\
 &\quad \left(\begin{bmatrix} B & \cdots & B \\ \vdots & \ddots & \vdots \\ B & \cdots & B \end{bmatrix} : \begin{bmatrix} d_{11} & \cdots & d_{1K} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{bmatrix} \right) \\
 &= \begin{bmatrix} (m_{11} + B) & \cdots & (m_{1K} + B) \\ \vdots & \ddots & \vdots \\ (m_{N1} + B) & \cdots & (m_{NK} + B) \end{bmatrix} : \begin{bmatrix} d_{11} & \cdots & d_{1K} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{bmatrix} \\
 &= \begin{bmatrix} \hat{m}_{11} & \cdots & \hat{m}_{1K} \\ \vdots & \ddots & \vdots \\ \hat{m}_{N1} & \cdots & \hat{m}_{NK} \end{bmatrix} : \begin{bmatrix} d_{11} & \cdots & d_{1K} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{bmatrix}
 \end{aligned}$$

where $:$ denotes the component-wise inner product of two matrices, $m_{ij} = \sum_{c=1}^C u_{ci}v_{cj}$, and $B = -\frac{\sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci}v_{cj}}{NK}$. It can be shown that we have $\sum_{i=1}^N \sum_{j=1}^K \hat{m}_{ij} = \sum_{i=1}^N \sum_{j=1}^K (m_{ij} + B) = 0$. This property of $\sum_{i=1}^N \sum_{j=1}^K \hat{m}_{ij} = \text{constant}$ is similar to the property $\sum_{i=1}^N \sum_{j=1}^K m_{ij} = \text{constant}$ in the FCCM model. We have explained in Section 3.3 how the latter property prevents the bias in the maximization of the degree of aggregation from occurring in the FCCM model. With the same reasoning, we have the maximization of the “transformed” degree of aggregation in FCR, i.e. the first and the second terms of J_{FCR} , free from the bias found in the DP-FCCM model. In the case of FCR, such a bias-free property is maintained regardless of the membership constraints imposed, i.e. $\sum_{i=1}^N \sum_{j=1}^K \hat{m}_{ij} = \text{constant}$ is always true for any constraint. Therefore, we can formulate a dual-partitioning-based fuzzy co-clustering algorithm by imposing the constraints in Eqs. (2.2) and (3.1) on the maximization of J_{FCR} .

The remaining terms, including the parameters T_u and T_v , in the J_{FCR} of FCR serve similar purposes as in the FCCM model (see Section 2.3.1 of Chapter 2).

By applying the Lagrange multiplier method, we can come up with the update membership equations in Eqs. (3.5) and (3.6). The algorithm FCR is simply a series of alternating updates of object and feature memberships by Eqs. (3.5) and (3.6) until J_{FCR} converges. Table 3.2 shows the pseudo-code of FCR. FCR has $O(CNK\tau)$ time complexity, which is the same as the time complexity of FCCM.

$$u_{ci} = \frac{\exp \left(\frac{\sum_{j=1}^K v_{cj}d_{ij} - \frac{\sum_{p=1}^N \sum_{j=1}^K d_{pj}}{NK} \sum_{j=1}^K v_{cj}}{T_u} \right)}{\sum_{f=1}^C \exp \left(\frac{\sum_{j=1}^K v_{fj}d_{ij} - \frac{\sum_{p=1}^N \sum_{j=1}^K d_{pj}}{NK} \sum_{j=1}^K v_{fj}}{T_u} \right)} \quad (3.5)$$

$$v_{cj} = \frac{\exp \left(\frac{\sum_{i=1}^N u_{ci} d_{ij} - \frac{\sum_{q=1}^K \sum_{i=1}^N d_{iq}}{NK} \sum_{i=1}^N u_{ci}}{T_v} \right)}{\sum_{f=1}^C \exp \left(\frac{\sum_{i=1}^N u_{fi} d_{ij} - \frac{\sum_{q=1}^K \sum_{i=1}^N d_{iq}}{NK} \sum_{i=1}^N u_{fi}}{T_v} \right)} \quad (3.6)$$

Table 3.2: FCR Pseudo-code

FCR ALGORITHM
<p>GIVEN: An $N \times K$ object-feature matrix (with $d_{ij} \geq 0$ as the element)</p>
<p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Set parameters T_u, T_v, C, τ_{max}, and ϵ; 2. Set $\tau = 0$; 3. Randomly initialize u_{ci}, $0 \leq u_{ci} \leq 1$; 4. REPEAT <ol style="list-style-type: none"> 5. Update v_{cj} by Eq. (3.6); 6. Update u_{ci} by Eq. (3.5); 7. Update $\tau = \tau + 1$; 8. UNTIL $\max_{c,i} u_{ci}(\tau) - u_{ci}(\tau - 1) \leq \epsilon$ or $\tau = \tau_{max}$;

The Concept Behind

We now explain the concept behind the “transformed” degree of aggregation (i.e. the first and the second terms of J_{FCR}), which is the key idea behind this model. Again, let us represent the transformed degree of aggregation as follows:

$$\begin{aligned}
 (\mathbf{M} : \mathbf{D}) - \mathbf{A}\mathbf{M} &= \left(\begin{bmatrix} m_{11} & \cdots & m_{1K} \\ \vdots & \ddots & \vdots \\ m_{N1} & \cdots & m_{NK} \end{bmatrix} : \begin{bmatrix} d_{11} & \cdots & d_{1K} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{bmatrix} \right) - \\
 &\quad \left(\begin{bmatrix} m_{11} & \cdots & m_{1K} \\ \vdots & \ddots & \vdots \\ m_{N1} & \cdots & m_{NK} \end{bmatrix} : \begin{bmatrix} A & \cdots & A \\ \vdots & \ddots & \vdots \\ A & \cdots & A \end{bmatrix} \right) \\
 &= \begin{bmatrix} m_{11} & \cdots & m_{1K} \\ \vdots & \ddots & \vdots \\ m_{N1} & \cdots & m_{NK} \end{bmatrix} : \begin{bmatrix} (d_{11} - A) & \cdots & (d_{1K} - A) \\ \vdots & \ddots & \vdots \\ (d_{N1} - A) & \cdots & (d_{NK} - A) \end{bmatrix}
 \end{aligned}$$

where $A = \frac{\sum_{p=1}^N \sum_{q=1}^K d_{pq}}{NK}$, i.e. the mean of the object-feature matrix. In this perspective, we can say that the purpose of the second term of J_{FCR} in Eq. (3.4) is to classify whether or not an object is considered related to a feature. An object x is considered related to a feature y if $d_{xy} > A$. In this case, a positive value in the corresponding m_{xy} will increase the objective function (see the expression above), bringing it closer to its local maximum. On the other hand, the same object and feature are considered not related if $d_{xy} < A$. Since we have $0 \leq m_{ij} \leq 1$ for $i = 1, \dots, N$ and $j = 1, \dots, K$, the algorithm will then try to assign a zero value to the corresponding m_{xy} (i.e. $m_{xy} = 0$) to prevent any decrease in the objective function. In FCR, the value of m_{ij} , from its definition $m_{ij} = \sum_{c=1}^C u_{ci} v_{cj}$, determines the membership distributions and consequently, the co-clusters that are generated by the algorithm. This explanation summarizes the basic principle followed by FCR in forming co-clusters.

The Proof of Convergence

We shall prove that the FCR algorithm in Table 3.2 always converges.

Theorem 1 *At every iteration τ , updating $v = \{v_{11}, \dots, v_{CK}\}$ by Eq. (3.6) never decreases the objective function J_{FCR} in Eq. (3.4).*

Proof. When updating v , we can consider the objective function J_{FCR} as a function of v alone, denoted by $J_{FCR}(v)$. This is because when updating v , we can consider all the u_{ci} terms in Eq. (3.4) as constants with values computed from the previous iteration. Theorem 1 can be proven by showing that the v^* (i.e v updated by Eq. (3.6)) is a local maximum of $J_{FCR}(v)$ in the region where the constraint in Eq. (3.1) is satisfied. From the fact that v^* is derived by applying the Lagrange multiplier method, we know that v^* has satisfied the first necessary condition and that it is a stationary point. But for it to be a local maximum, we need to show that the Hessian matrix $\nabla^2 J_{FCR}(v^*)$ is negative definite. It can be shown that:

$$\nabla^2 J_{FCR}(v) = \begin{bmatrix} \frac{\partial J_{FCR}(v)^2}{\partial v_{11}v_{11}} & \dots & \frac{\partial J_{FCR}(v)^2}{\partial v_{11}v_{CK}} \\ \vdots & \ddots & \vdots \\ \frac{\partial J_{FCR}(v)^2}{\partial v_{CK}v_{11}} & \dots & \frac{\partial J_{FCR}(v)^2}{\partial v_{CK}v_{CK}} \end{bmatrix} = \begin{bmatrix} -\frac{T_v}{v_{11}} & & 0 \\ & \ddots & \\ 0 & & -\frac{T_v}{v_{CK}} \end{bmatrix}$$

At v^* , we have $v_{cj} \geq 0$ for all $c = 1, \dots, C$ and $j = 1 \dots, K$ (by Eq. (3.6)). And we always assign a positive value to T_v . Therefore, the Hessian matrix $\nabla^2 J_{FCR}(v^*)$ is negative definite. Since both the first necessary condition and the second sufficient condition ($\nabla^2 J_{FCR}(v^*)$ is negative definite) are satisfied, v^* updated from Eq. (3.6) is indeed a local maximum of $J_{FCR}(v)$ in the region where the constraint in Eq. (3.1) is satisfied. For this reason, at every iteration, updating v by Eq. (3.6) never decreases the objective function J_{FCR} . ■

Theorem 2 *At every iteration τ , updating $u = \{u_{11}, \dots, u_{CN}\}$ by Eq. (3.5) never decreases the objective function J_{FCR} in Eq. (3.4).*

Proof. Theorem 2 can be proven in a similar fashion as Theorem 1. ■

Theorem 3 *The following is true, with J_{FCR} is as defined in Eq. (3.4).*

$$J_{FCR} \leq \sum_{i=1}^N \sum_{j=1}^K d_{ij} - T_u N \ln \left(\frac{1}{C} \right) - T_v K \ln \left(\frac{1}{C} \right)$$

Proof. Since $0 \leq u_{ci}, v_{cj} \leq 1$ (by Eqs. (3.5) and (3.6)), it can be shown that the upper limit of the summation of the first two terms of J_{FCR} in Eq. (3.4) is loosely bounded by $\sum_{i=1}^N \sum_{j=1}^K d_{ij}$:

$$\left(\sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj} d_{ij} - \frac{\sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj}}{NK} \sum_{i=1}^N \sum_{j=1}^K d_{ij} \right) \leq \sum_{i=1}^N \sum_{j=1}^K d_{ij}$$

Also, the fuzzifier terms of J_{FCR} are maximized when $u_{ci} = \frac{1}{C}$ and $v_{cj} = \frac{1}{C}$ for $c = 1, \dots, C$, $i = 1, \dots, N$, and $j = 1, \dots, K$, which corresponds to the maximum entropy condition [22]. Considering the constraints in Eqs. (2.2) and (3.1), this translates into the following:

$$-T_u \sum_{c=1}^C \sum_{i=1}^N u_{ci} \ln u_{ci} - T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj} \ln v_{cj} \leq -T_u N \ln \left(\frac{1}{C} \right) - T_v K \ln \left(\frac{1}{C} \right)$$

Thus we have the proof of Theorem 3. ■

Corollary 1 *The FCR algorithm in Table 3.2 converges to a local maximum of the optimization.*

Proof. This corollary is a direct consequence of the three theorems above. Theorems 1 and 2 indicate that the procedure in Table 3.2 never decreases the FCR objective function. Theorem 3 states that there is a limit to how much this objective function can be increased. So eventually the procedure should stop somewhere before or when it reaches this limit. ■

In summary, we have proven that under the updating rules of Eqs. (3.5) and (3.6) the algorithm FCR converges to a local maximum of the objective function in Eq. (3.4) in the space constrained by Eqs. (2.2) and (3.1).

An Illustration

In this section, we provide an illustration how FCR performs on a simple artificial dataset D_1 in Figure 3.1, with rows and columns correspond to objects and features respectively.

$$D_1 = \begin{bmatrix} \begin{bmatrix} 1 & 1 & 0.2 \\ 0.2 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 \end{bmatrix} \\ \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 0.2 \\ 0.2 & 1 & 1 \end{bmatrix} \end{bmatrix}$$

Figure 3.1: An Example D_1

The two rectangles denote the two ideal co-clusters. These ideal co-clusters correspond to the following object (U) and feature (V) membership matrices.

$$U = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad V = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Figure 3.2: FCR Simplified Memberships Search Spaces

We intend to present the 2-D plots of the simplified membership search spaces of the DP-FCCM model and FCR for the given D_1 . By doing so, we can see how the global maximum is shifted from the location that corresponds to the generation of a single co-clustering in the DP-FCCM model, to the location that corresponds to the correct clustering in FCR. To draw these plots, we assume that v_{cj} is crisp (i.e. 0 or 1). Then, we generate all possible combinations of the feature membership matrix V . These combinations are finite because the membership is crisp and it must conform to the constraint in Eq. (3.1). For each of these combinations, we compute the corresponding u_{ci} using Eq. (3.5) in the case of FCR, and Eq. (3.2) in the case of DP-FCCM (the latter equation is shown in Section 3.4.2, and it is derived by applying the Lagrange multiplier method on the DP-FCCM model). Based on these v_{cj} and u_{ci} , we calculate the objective functions J_{FCCM} for the DP-FCCM model and J_{FCR} for FCR. Figure 3.2 shows the 2-D plots for these two objective functions.

The arrows in the figures indicate the locations of the global maxima. In the case of J_{FCCM} , the two arrows correspond to the object and feature membership matrices as follows, with rows and columns representing co-clusters and objects or features respectively:

$$\begin{aligned} U_{FCCM} &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} & V_{FCCM} &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ U_{FCCM} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} & V_{FCCM} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{aligned}$$

In both the global maxima of J_{FCCM} in the DP-FCCM model, all objects and features are assigned to the same single co-cluster. In the case of J_{FCR} , the two arrows correspond to the object and feature membership matrices, which are exactly the same as the ideal co-clusters depicted above.

$$\begin{aligned} U_{\text{FCR}} &= \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} & V_{\text{FCR}} &= \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \\ U_{\text{FCR}} &= \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} & V_{\text{FCR}} &= \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

3.4.2 The Second Model: The HFCR Algorithm

The Algorithm

Let us revisit the DP-FCCM model in Section 3.3, specifically focusing on its update membership equations in Eqs. (3.2) and (3.3). By comparing Eqs. (3.2) and (3.3) with update membership equations of FCR in Eqs. (3.5) and (3.6), we can get a hint why the former equations do not work (i.e. they lead to the generation of a single co-cluster), while the latter equations do. Let us consider Eq. (3.2). The fundamental component of this equation lies in the term $\exp\left(\frac{\sum_{j=1}^K v_{cj} d_{ij}}{T_u}\right)$, which measures the similarity between an object i and a co-cluster c . It can be seen however that, since in the DP-FCCM model we have $\sum_{j=1}^K v_{cj} \neq \text{constant}$, the similarity measure $\exp\left(\frac{\sum_{j=1}^K v_{cj} d_{ij}}{T_u}\right)$ favors co-clusters with larger $\sum_{j=1}^K v_{cj}$ values (i.e. such co-clusters tend to be assigned larger membership values). In other words, it favors co-clusters that contain more features. Contrast this with the term $\exp\left(\frac{\sum_{j=1}^K v_{cj} d_{ij} - \frac{\sum_{p=1}^N \sum_{j=1}^K d_{pj}}{NK} \sum_{j=1}^K v_{cj}}{T_u}\right)$ of Eq. (3.5) in FCR. The term $-\frac{\sum_{p=1}^N \sum_{j=1}^K d_{pj}}{NK} \sum_{j=1}^K v_{cj}$ in this equation provides a negative force opposing the bias toward a larger $\sum_{j=1}^K v_{cj}$ value. In a similar fashion, we can show that the update feature membership equation in Eq. (3.3) is biased towards co-clusters with larger $\sum_{i=1}^N u_{ci}$ values (i.e. towards co-clusters containing more objects); while in the case of FCR, the term $-\frac{\sum_{q=1}^K \sum_{i=1}^N d_{iq}}{NK} \sum_{i=1}^N u_{ci}$ in Eq. (3.6) provides a negative force opposing this bias. In Section 3.4.1, we have seen how the second term in J_{FCR} in Eq. (3.4) can remove the bias in the maximization of the degree of aggregation. Here we can see the reflection of such a mechanism in the corresponding update membership

equations.

This understanding motivates us to formulate an alternative fuzzy co-clustering algorithm based on the dual-partitioning approach. We have seen that the dual-partitioning approach cannot be accomplished if there are biases in the update equations, such as the ones found in Eqs. (3.2) and (3.3). FCR, through its objective function, provides one mechanism to counter such biases in the update equations. Now, we formulate another mechanism based on a heuristic, in which the biases are removed directly from the update membership equations. Consider the following update equations:

$$u_{ci} = \frac{\exp \left\{ \frac{\sum_{j=1}^K v_{cj} d_{ij}}{T_u \sum_{j=1}^K v_{cj}} \right\}}{\sum_{f=1}^C \exp \left\{ \frac{\sum_{j=1}^K v_{fj} d_{ij}}{T_u \sum_{j=1}^K v_{fj}} \right\}} \quad (3.7)$$

$$v_{cj} = \frac{\exp \left\{ \frac{\sum_{i=1}^N u_{ci} d_{ij}}{T_v \sum_{i=1}^N u_{ci}} \right\}}{\sum_{f=1}^C \exp \left\{ \frac{\sum_{i=1}^N u_{fi} d_{ij}}{T_v \sum_{i=1}^N u_{fi}} \right\}} \quad (3.8)$$

We can see that in Eq. (3.7), the argument of the exponential term $\exp \left(\frac{\sum_{j=1}^K v_{cj} d_{ij}}{T_u} \right)$ in Eq. (3.2) is normalized by $\sum_{j=1}^K v_{cj}$, removing the bias that favors co-clusters with larger $\sum_{j=1}^K v_{cj}$ values. The same thing can be said for Eq. (3.8). This is the key idea behind the proposed algorithm Heuristic Fuzzy Co-clustering with the Ruspini's condition (HFCR) [124]. The HFCR algorithm becomes a series of alternative updates between Eqs. (3.7) and (3.8), until all the memberships converge to their stable values. Except for the update membership equations used, the pseudo-code of HFCR shown in Table 3.3 is exactly the same as the pseudo-code of FCR in Table 3.2. HFCR has the same time complexity as FCR, i.e. $O(CNK\tau)$.

Table 3.3: HFCR Pseudo-code

HFCR ALGORITHM
<p>GIVEN: An $N \times K$ object-feature matrix (with $d_{ij} \geq 0$ as the element)</p>
<p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Set parameters T_u, T_v, C, τ_{max}, and ϵ; 2. Set $\tau = 0$; 3. Randomly initialize u_{ci}, $0 \leq u_{ci} \leq 1$; 4. REPEAT <ol style="list-style-type: none"> 5. Update v_{cj} by Eq. (3.8); 6. Update u_{ci} by Eq. (3.7); 7. Update $\tau = \tau + 1$; 8. UNTIL $\max_{c,i} u_{ci}(\tau) - u_{ci}(\tau - 1) \leq \epsilon$ or $\tau = \tau_{max}$;

Some Conceptual Justifications

In the case of FCR, our understanding of its objective function provides a justification to the validity of the algorithm. In the case of HFCR however, given that it is heuristically formulated, no explicit corresponding objective function is present. Thus, we cannot derive a justification for HFCR from there. To accommodate the justification of the heuristic-based algorithm, let us propose two auxiliary functions in Eqs. (3.9) and (3.10).

$$\begin{aligned}
 J_1 = & \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} \frac{v_{cj}}{\sum_{q=1}^K v_{cq}} d_{ij} - T_u \sum_{c=1}^C \sum_{i=1}^N u_{ci} \ln u_{ci} - \\
 & T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj} \ln v_{cj}
 \end{aligned} \tag{3.9}$$

$$\begin{aligned}
 J_2 = & \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K \frac{u_{ci}}{\sum_{p=1}^N u_{cp}} v_{cj} d_{ij} - T_u \sum_{c=1}^C \sum_{i=1}^N u_{ci} \ln u_{ci} - \\
 & T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj} \ln v_{cj}
 \end{aligned} \tag{3.10}$$

Considering their structures, both J_1 and J_2 share similar principles with J_{FCCM} in Eq. (2.11), that is, the maximization of these three functions are intended to make highly-related objects and features (i.e. those with high d_{ij} values) in a dataset to be co-clustered together, resulting in highly-coherent co-clusters. The three however differ in how normalizations are applied on their respective degree of aggregation terms (i.e. the first term of each objective function). In J_{FCCM} , no normalization takes place in its degree of aggregation term. In J_1 however, the degree of aggregation of every co-cluster c is normalized by $\sum_{q=1}^K v_{cq}$, while in J_2 , it is normalized by $\sum_{p=1}^N u_{cp}$. These normalizations are essential to avoid the bias in the maximization of the degree of aggregation (e.g. the bias that occurs in the DP-FCCM model as explained in Section 3.3). Let us denote the first terms of J_1 and J_2 respectively by DoA_1 and DoA_2 . We can represent them as follows:

$$\begin{aligned}
 DoA_1 &= \begin{bmatrix} (m_1)_{11} & \cdots & (m_1)_{1K} \\ \vdots & \ddots & \vdots \\ (m_1)_{N1} & \cdots & (m_1)_{NK} \end{bmatrix} : \begin{bmatrix} d_{11} & \cdots & d_{1K} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{bmatrix} \\
 DoA_2 &= \begin{bmatrix} (m_2)_{11} & \cdots & (m_2)_{1K} \\ \vdots & \ddots & \vdots \\ (m_2)_{N1} & \cdots & (m_2)_{NK} \end{bmatrix} : \begin{bmatrix} d_{11} & \cdots & d_{1K} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{bmatrix}
 \end{aligned}$$

Here we have $(m_1)_{ij} = \sum_{c=1}^C u_{ci} \frac{v_{cj}}{\sum_{q=1}^K v_{cq}}$ and $(m_2)_{ij} = \sum_{c=1}^C \frac{u_{ci}}{\sum_{p=1}^N u_{cp}} v_{cj}$. By observing the constraints in Eqs. (2.2) and (3.1), it should not be difficult to establish that $\sum_{i=1}^N \sum_{j=1}^K (m_1)_{ij} = N$ and $\sum_{i=1}^N \sum_{j=1}^K (m_2)_{ij} = K$. By eliminating the variations in the values $\sum_{i=1}^N \sum_{j=1}^K (m_1)_{ij}$ and $\sum_{i=1}^N \sum_{j=1}^K (m_2)_{ij}$, just like in the FCR and FCCM models, we remove the biases in J_1 and J_2 respectively.

It can be shown, by applying the Lagrange multiplier method, that the update equations in Eqs. (3.7) and (3.8) can respectively be derived from $\frac{\partial L_1}{\partial u_{ci}} = 0$ and $\frac{\partial L_2}{\partial v_{cj}} = 0$, where L_1 and L_2 are the Lagrangian functions of J_1 and J_2 respectively taking into account the constraints in Eqs. (2.2) and (3.1). It can be shown that the Hessian matrix $\nabla^2 J_1(u)$ is negative definite if u_{ci} is updated by Eq. (3.7). This indicates that for a given $v = [v_{11}, v_{12}, \dots, v_{CK}]^T$, a u_{ci} update in HFCR always points to a local maximum of J_1 in the region that satisfies the constraints in Eqs. (2.2) and (3.1), and thus the update always seeks to increase the J_1 function. As we mentioned earlier, from the structure of the function, an increase in J_1 , similar to an increase in J_{FCCM} , promotes the formations of highly coherent co-clusters by assigning highly related objects and features into the same co-clusters. In a similar fashion we can prove that for a given $u = [u_{11}, u_{12}, \dots, u_{CN}]^T$, a v_{cj} update in HFCR always points to a local maximum of J_2 in the region that satisfies the constraints in Eqs. (2.2) and (3.1), and thus the update always seeks to increase the J_2 function. Again from the definition of J_2 , an increase in the function promotes the formations of highly coherent co-clusters just like in the case of J_{FCCM} . Based on this understanding, we can justify that the heuristic-natured updates in HFCR direct the algorithm to a proper aim of generating highly coherent co-clusters. This is further confirmed by our experimental studies of HFCR on large datasets in [95] and Section 3.6.

3.4.3 The Third Model: The IPFC Algorithm

The Algorithm

We have discussed in Section 3.4.1 that the purpose of the second term of J_{FCR} in Eq. (3.4) can be seen as to classify whether or not an object is considered related to a feature. There we see, whether an object i is considered related or not to a feature j depends on the value of d_{xy} relative to $A = \frac{\sum_{p=1}^N \sum_{q=1}^K d_{pq}}{NK}$, i.e. the mean of the object-feature matrix (see the explanation in Section 3.4.1). Thus, A plays a critical role in FCR. While in an ideal case A alone can be a good option to act as a central point separating related object-feature pairs (high d_{xy}) from the non-related ones (low d_{xy}), in reality, due to the presence of noise in a dataset, high-dependency on A like what we have in FCR may become prohibitive. It is a well-known fact that a mean value is not robust in the presence of noise [109]. Because of this, the application of FCR is limited to simple applications with limited amount of noise presents. We shall demonstrate this limitation of FCR in Section 3.6.

In this section we propose the Improved Partitioning-based Fuzzy Co-clustering (IPFC) algorithm, whose formulation can be regarded as one possible way to improve the robustness of FCR in the presence of noise [125]. The goal of IPFC is to maximize the objective function given in Eq. (3.11), subject to the constraints given in Eqs. (2.2) and (3.1).

$$\begin{aligned}
 J_{IPFC} = & 2 \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj} d_{ij} - \sum_{i=1}^N \left(\frac{\sum_{c=1}^C \sum_{j=1}^K u_{ci} v_{cj}}{K} \sum_{j=1}^K d_{ij} \right) - \\
 & \sum_{j=1}^K \left(\frac{\sum_{c=1}^C \sum_{i=1}^N u_{ci} v_{cj}}{N} \sum_{i=1}^N d_{ij} \right) - T_u \sum_{c=1}^C \sum_{i=1}^N u_{ci} \ln u_{ci} - \\
 & T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj} \ln v_{cj}
 \end{aligned} \tag{3.11}$$

The first three terms of the objective function J_{IPFC} (i.e. excluding the fuzzifier terms) can be represented as follows:

$$\begin{aligned}
 & \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj} d_{ij} - \sum_{i=1}^N \left(\frac{\sum_{c=1}^C \sum_{j=1}^K u_{ci} v_{cj}}{K} \sum_{j=1}^K d_{ij} \right) + \\
 & \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj} d_{ij} - \sum_{j=1}^K \left(\frac{\sum_{c=1}^C \sum_{i=1}^N u_{ci} v_{cj}}{N} \sum_{i=1}^N d_{ij} \right) = \\
 & \overbrace{\sum_{i=1}^N \left(\begin{bmatrix} m_{i1} \\ \vdots \\ m_{iK} \end{bmatrix}^T \cdot \begin{bmatrix} (d_{i1} - A_i^{row}) \\ \vdots \\ (d_{iK} - A_i^{row}) \end{bmatrix} \right)}^l + \overbrace{\sum_{j=1}^K \left(\begin{bmatrix} m_{1j} \\ \vdots \\ m_{Nj} \end{bmatrix}^T \cdot \begin{bmatrix} (d_{1j} - A_j^{col}) \\ \vdots \\ (d_{Nj} - A_j^{col}) \end{bmatrix} \right)}^h
 \end{aligned}$$

where $A_i^{row} = \frac{\sum_{j=1}^K d_{ij}}{K}$ for $i = 1, \dots, N$ and $A_j^{col} = \frac{\sum_{i=1}^N d_{ij}}{N}$ for $j = 1, \dots, K$, i.e. the mean of each row and the mean of each column respectively. In the term l above, the new algorithm emulates the same process of distinguishing related object-feature pairs (high d_{ij}) from non-related object-feature pairs (low d_{ij}) in FCR, as discussed in Section 3.4.1. In this case however, the process is limited only to the row of the object-feature matrix (i.e. the object vector). The same can be said for the term h , albeit this time, the process is limited only to the column of the object-feature matrix (i.e. the feature

vector). Compared to FCR, this new model offers three advantages: 1) The reduction of the noise's impact by limiting their scope to the row and column (vectors) levels, as opposed to the whole dataset (matrix) level as in the case of FCR. Since the system can now make the decision to consider whether or not an object-feature pair is related at the row and column levels based on their respective mean values A_i^{row} and A_j^{col} , it no longer needs to rely only on the matrix mean, i.e. $A = \frac{\sum_{p=1}^N \sum_{q=1}^K d_{pq}}{NK}$, whose value may have been contaminated by the aggregated impact of all the noise in a dataset.

2) IPFC offers a more rigorous scheme for the construction of co-clusters. In IPFC, in order to consider whether or not an object x is related to feature y , it needs to consult two entities: A_x^{row} and A_y^{col} . In other words, the algorithm must check the position of this pair (d_{xy}) relative to the positions of the other pairs along the same row (d_{iy} for $i = 1, \dots, N$) and the positions of the other pairs along the same column (d_{xj} for $j = 1, \dots, K$). This scheme allows the system to double-check the accuracy of its judgement; something that is not present in FCR.

3) The objective function of IPFC adopts a similar mechanism to the one FCR to deal with the bias in the maximization of the degree of aggregation as explained in Section 3.3. Again, the first three terms in J_{IPFC} can be represented as follows:

$$\begin{aligned}
 & \overbrace{\sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj} d_{ij} - \sum_{i=1}^N \left(\frac{\sum_{c=1}^C \sum_{j=1}^K u_{ci} v_{cj}}{K} \sum_{j=1}^K d_{ij} \right)}^r + \\
 & \overbrace{\sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj} d_{ij} - \sum_{j=1}^K \left(\frac{\sum_{c=1}^C \sum_{i=1}^N u_{ci} v_{cj}}{N} \sum_{i=1}^N d_{ij} \right)}^s = \\
 & \underbrace{\begin{bmatrix} (m_{11} + B_1^{row}) & \cdots & (m_{1K} + B_1^{row}) \\ \vdots & \ddots & \vdots \\ (m_{N1} + B_N^{row}) & \cdots & (m_{NK} + B_N^{row}) \end{bmatrix}}_r : \begin{bmatrix} d_{11} & \cdots & d_{1K} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{bmatrix} + \\
 & \underbrace{\begin{bmatrix} (m_{11} + B_1^{col}) & \cdots & (m_{1K} + B_K^{col}) \\ \vdots & \ddots & \vdots \\ (m_{N1} + B_1^{col}) & \cdots & (m_{NK} + B_K^{col}) \end{bmatrix}}_s : \begin{bmatrix} d_{11} & \cdots & d_{1K} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{bmatrix}
 \end{aligned}$$

where $B_i^{row} = -\frac{\sum_{c=1}^C \sum_{q=1}^K u_{ci} v_{cq}}{K}$ for $i = 1, \dots, N$, and $B_j^{col} = -\frac{\sum_{c=1}^C \sum_{p=1}^N u_{cp} v_{cj}}{N}$ for $j = 1, \dots, K$. It can be shown that $\sum_{i=1}^N \sum_{j=1}^K (m_{ij} + B_i^{row}) = \text{constant} = 0$ and $\sum_{i=1}^N \sum_{j=1}^K (m_{ij} + B_j^{col}) = \text{constant} = 0$. With these two equalities, we emulate the $\sum_{i=1}^N \sum_{j=1}^K m_{ij} = \text{constant}$ property of the existing FCCM model to prevent any bias in the maximization of the terms r and s above respectively. Note that we employ a similar strategy in FCR as explained in Section 3.4.1.

Using the Lagrange Multiplier method, we can derive the update membership equations in Eqs. (3.12) and (3.13) for IPFC. The algorithm proceeds by updating Eqs. (3.12) and (3.13) in an alternating fashion until the objective function converges. Except for the update membership equations used, the pseudo-code of IPFC in Table 3.4 is exactly the same as the pseudo-code of FCR in Table 3.2. The time complexity of

IPFC is $O(CNK\tau)$.

$$u_{ci} = \frac{\exp \left[\frac{1}{T_u} \left\{ 2 \sum_{j=1}^K v_{cj} d_{ij} - \frac{\sum_{j=1}^K v_{cj}}{K} \sum_{j=1}^K d_{ij} - \sum_{j=1}^K \left(\frac{v_{cj}}{N} \sum_{p=1}^N d_{pj} \right) \right\} \right]}{\sum_{e=1}^C \exp \left[\frac{1}{T_u} \left\{ 2 \sum_{j=1}^K v_{ej} d_{ij} - \frac{\sum_{j=1}^K v_{ej}}{K} \sum_{j=1}^K d_{ij} - \sum_{j=1}^K \left(\frac{v_{ej}}{N} \sum_{p=1}^N d_{pj} \right) \right\} \right]} \quad (3.12)$$

$$v_{cj} = \frac{\exp \left[\frac{1}{T_v} \left\{ 2 \sum_{i=1}^N u_{ci} d_{ij} - \frac{\sum_{i=1}^N u_{ci}}{N} \sum_{i=1}^N d_{ij} - \sum_{i=1}^N \left(\frac{u_{ci}}{K} \sum_{q=1}^K d_{iq} \right) \right\} \right]}{\sum_{e=1}^C \exp \left[\frac{1}{T_v} \left\{ 2 \sum_{i=1}^N u_{ei} d_{ij} - \frac{\sum_{i=1}^N u_{ei}}{N} \sum_{i=1}^N d_{ij} - \sum_{i=1}^N \left(\frac{u_{ei}}{K} \sum_{q=1}^K d_{iq} \right) \right\} \right]} \quad (3.13)$$

Table 3.4: IPFC Pseudo-code

IPFC ALGORITHM
GIVEN: An $N \times K$ object-feature matrix (with $d_{ij} \geq 0$ as the element)
PROCEDURE: 1. Set parameters T_u , T_v , C , τ_{max} , and ϵ ; 2. Set $\tau = 0$; 3. Randomly initialize u_{ci} , $0 \leq u_{ci} \leq 1$; 4. REPEAT 5. Update v_{cj} by Eq. (3.13); 6. Update u_{ci} by Eq. (3.12); 7. Update $\tau = \tau + 1$; 8. UNTIL $\max_{c,i} u_{ci}(\tau) - u_{ci}(\tau - 1) \leq \epsilon$ or $\tau = \tau_{max}$;

The Proof of Convergence

We shall prove that the IPFC algorithm in Table 3.4 always converges.

Theorem 4 *At every iteration τ , updating $v = \{v_{11}, \dots, v_{CK}\}$ by Eq. (3.13) never decreases the objective function J_{IPFC} in Eq. (3.11).*

Proof. When updating v , we can consider the objective function J_{IPFC} as a function of v alone, denoted by $J_{IPFC}(v)$. This is because when updating v , we can consider all the u_{ci} terms in Eq. (3.11) as constants with values computed from the previous iteration. Theorem 4 can be proven by showing that the v^* (i.e v updated by Eq. (3.13)) is a local maximum of $J_{IPFC}(v)$ in the region where the constraint in Eq. (3.1) is satisfied. From the fact that v^* is derived by applying the Lagrange multiplier method, we know that v^* has satisfied the first necessary condition and that it is a stationary point. But for it to be a local maximum, we need to show that the Hessian matrix $\nabla^2 J_{IPFC}(v^*)$ is negative definite. It can be shown that:

$$\nabla^2 J_{IPFC}(v) = \begin{bmatrix} \frac{\partial J_{IPFC}(v)^2}{\partial v_{11}v_{11}} & \dots & \frac{\partial J_{IPFC}(v)^2}{\partial v_{11}v_{CK}} \\ \vdots & \ddots & \vdots \\ \frac{\partial J_{IPFC}(v)^2}{\partial v_{CK}v_{11}} & \dots & \frac{\partial J_{IPFC}(v)^2}{\partial v_{CK}v_{CK}} \end{bmatrix} = \begin{bmatrix} -\frac{T_v}{v_{11}} & & 0 \\ & \ddots & \\ 0 & & -\frac{T_v}{v_{CK}} \end{bmatrix}$$

At v^* , we have $v_{cj} \geq 0$ for all $c = 1, \dots, C$ and $j = 1, \dots, K$ (by Eq. (3.13)). And we always assign a positive value to T_v . Therefore, the Hessian matrix $\nabla^2 J_{IPFC}(v^*)$ is negative definite. Since both the first necessary condition and the second sufficient condition ($\nabla^2 J_{IPFC}(v^*)$ is negative definite) are satisfied, v^* updated from Eq. (3.13) is indeed a local maximum of $J_{IPFC}(v)$ in the region where the constraint in Eq. (3.1) is satisfied. For this reason, at every iteration, updating v by Eq. (3.13) never decreases the objective function J_{IPFC} . ■

Theorem 5 *At every iteration τ , updating $u = \{u_{11}, \dots, u_{CN}\}$ by Eq. (3.12) never decreases the objective function J_{FCR} in Eq. (3.4).*

Proof. Theorem 5 can be proven in a similar fashion as Theorem 4. ■

Theorem 6 *The following is true, with J_{IPFC} is as defined in Eq. (3.11).*

$$J_{IPFC} \leq 2 \sum_{i=1}^N \sum_{j=1}^K d_{ij} - T_u N \ln \left(\frac{1}{C} \right) - T_v K \ln \left(\frac{1}{C} \right)$$

Proof. Since $0 \leq u_{ci}, v_{cj} \leq 1$ (from Eqs. (3.12) and (3.13)), we have the upper limit of the first three terms of Eq. (3.11) loosely bounded by $2 \sum_{i=1}^N \sum_{j=1}^K d_{ij}$:

$$\begin{aligned} 2 \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj} d_{ij} &= \sum_{i=1}^N \left(\frac{\sum_{c=1}^C \sum_{j=1}^K u_{ci} v_{cj}}{K} \sum_{j=1}^K d_{ij} \right) - \\ &\quad \sum_{j=1}^K \left(\frac{\sum_{c=1}^C \sum_{i=1}^N u_{ci} v_{cj}}{N} \sum_{i=1}^N d_{ij} \right) \leq 2 \sum_{i=1}^N \sum_{j=1}^K d_{ij} \end{aligned}$$

Also, the fuzzifier terms of J_{IPFC} are maximized when $u_{ci} = \frac{1}{C}$ and $v_{cj} = \frac{1}{C}$ for $c = 1, \dots, C$, $i = 1, \dots, N$, and $j = 1, \dots, K$, which corresponds to the maximum entropy condition [22]. Considering the constraints in Eqs. (2.2) and (3.1), this translates into the following:

$$-T_u \sum_{c=1}^C \sum_{i=1}^N u_{ci} \ln u_{ci} - T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj} \ln v_{cj} \leq -T_u N \ln \left(\frac{1}{C} \right) - T_v K \ln \left(\frac{1}{C} \right)$$

Thus we have the proof of Theorem 6. ■

Corollary 2 *The IPFC algorithm converges to a local maximum of the optimization.*

Proof. This corollary is a direct consequence of the three theorems above. Theorems 4 and 5 indicate that the iterative membership-update procedure in IPFC never decreases the objective function in Eq. (3.11). Theorem 6 states that there is a limit to how

much this objective function can be increased. So eventually the procedure should stop somewhere before or when it reaches this limit. ■

In summary, we have proven that under the updating rules of Eqs. (3.12) and (3.13) the algorithm IPFC converges to a local maximum of the objective function in Eq. (3.11) in the space constrained by Eqs. (2.2) and (3.1).

3.5 Advantages and Disadvantages of the Dual-partitioning Approach

As discussed in Section 3.2, the key difference between the dual-partitioning approach and the existing FCCM model lies in the constraints imposed on the feature memberships. This difference leads to some advantages and disadvantages in the two techniques. In this section, we look at some of the key advantages and disadvantages. These explanations are useful to support the motivation behind the proposed fuzzy-possibilistic co-clustering framework, which is discussed in Chapters 4 and 5.

Let us first repeat the explanations on the two different types of feature memberships in the two fuzzy co-clustering approaches. The dual-partitioning approach has a feature partitioning membership. In this case, the membership of a feature j in a cluster (or co-cluster) c (i.e. v_{cj}) is decided based on the similarity between j and c , relative to the similarities of j and all the other clusters existing in a dataset. In all FCR, HFCR, and IPFC, the former similarity measure is captured by the numerator portions of their respective v_{cj} update equations (Eqs. (3.6), (3.8), and (3.13)), while the latter similarity measure is captured by the corresponding denominator portions. This “relative to other clusters” notion causes the feature membership to capture the partitioning structure of features [126], because the membership reflects which clusters, out of all the clusters in a

dataset, the features should belong to. On the other hand, the existing FCCM model has a feature ranking membership. In this case, the membership of a feature j in a cluster c is decided based on the similarity between j and c , relative to the similarities between all the other features and c . In FCCM, the former similarity measure is captured by the numerator portion of its v_{cj} update equation (Eq. (2.14)), while the latter similarity measure is captured by the corresponding denominator portion. This “relative to other features” causes the feature membership to capture the ranking structure of features, because the membership reflects which features, out of all the features in a dataset, are the more important ones in clusters.

Figure 3.3: A Common Clustering Situation 1

To get a better understanding on the difference between partitioning and ranking memberships, let us go through a simple example of computing the two memberships in a common clustering situation reflected in Figure 3.3. In the figure, we have 3 clusters: C_1 , C_2 , and C_3 . Consider the point P located near the center of C_1 . Intuitively, this point should have a membership in C_1 greater than in C_2 , which in turn greater than in

C_3 , i.e. $v_{C_1P} > v_{C_2P} > v_{C_3P}$. We intend to see if partitioning and ranking memberships would agree on this in the conceptual level. In the partitioning membership case, compared to C_2 or C_3 (i.e. the notion of relative to other clusters), P is closer to C_1 . Therefore, P would have a higher partitioning membership in C_1 than in the other two clusters. Also, since relative to C_3 , P is closer to C_2 , we should have the partitioning membership of P in C_2 greater than in C_3 . Thus, we should have the partitioning membership of P as follows: $v_{C_1P} > v_{C_2P} > v_{C_3P}$, which agrees with our intuition. In the ranking membership case, P would have a high membership to C_1 because compared to most other points in the dataset (i.e. the notion of relative to other points/features), it has a relatively closer distance to (the center of) C_1 . In C_2 however, P would have a medium ranking membership because there are many points that, compared to P , are closer to C_2 , (i.e. the points in the region denoted by Rect1 in the figure), but at the same time, there are also many other points that, compared to P , are further away from C_2 (i.e. the points in the region denoted by Rect2). In C_3 , P would have a low ranking membership because most other points in the dataset have closer distances to C_3 . The resulting ranking membership of P therefore follows $v_{C_1P} > v_{C_2P} > v_{C_3P}$, which is also consistent with our intuition, just like in the partitioning case.

3.5.1 Advantages

Consider Figure 3.4, which depicts the same dataset and clusters as the ones in Figure 3.3. We note that the three clusters have some overlapping regions. Let us this time consider the point Q located in the overlapping region between C_1 and C_2 . For clarity sake, let us consider an extreme case where the distance from Q to C_1 is equal to the distance from Q to C_2 . Intuitively, we expect Q to have an equal membership value in C_1 and in C_2 , and a relatively lower membership value in C_3 ; or in other words

Figure 3.4: A Common Clustering Situation 2

$v_{1Q} = v_{2Q} > v_{3Q}$. In the partitioning membership case, compared to C_2 , Q is equally distanced from C_1 ; while compared to C_3 , Q is closer to C_1 . Therefore, Q would have a membership $v_{1Q} = v_{2Q} > v_{3Q}$, which is consistent with our intuition. In the ranking membership case, Q would have a quite low ranking membership in C_3 because there are a significant number of other points positioned closer to C_3 than Q . In C_1 , Q would have a relatively higher-ranking membership than in C_3 because there are a significant number of other points positioned further away from C_1 than Q . In C_2 , compared to Q , the number of points positioned further away from C_2 is approximately the same as the number of points positioned further away from C_1 . However, on aggregate, these points have closer distances to C_2 than to C_1 (e.g. the points in the region denoted by Rect3 are all closer to C_2 than to C_1). For this reason, Q would have a lower ranking membership in C_2 than in C_1 . Thus we have the resulting ranking membership of Q : $v_{1Q} > v_{2Q} > v_{3Q}$, which does not agree with our intuition. Therefore, by relying on a ranking membership, it may look like Q is more important in C_1 than in C_2 , although

the figure actually indicates that it should be equally important in both clusters. In the case of a ranking membership, such a “misrepresentation” may occur when evaluating points in overlapping region. It can be shown that misrepresentation of the same nature can also occur if we consider the ranking membership of point R in Figure 3.4, which is located in the overlapping region as well, but can be considered to have a less extreme position than the position of Q (i.e. R is closer to C_2 than to C_1).

One contributing factor to this misrepresentation is that the determination of a ranking membership of a point in a particular cluster ignores the position of the point in all the other clusters (i.e. the absence of the relative to other clusters notion, as found in a partitioning membership). So as much as a ranking membership is concerned, the presence of other clusters is, to a large degree, ignored, and therefore, the notion of overlaps with other clusters is not really captured. This factor, added by another factor that the determination of a ranking membership takes into account the relative position of a point in a cluster to all the other points can, as demonstrated earlier, lead to the misrepresentation of memberships of points in regions where clusters overlap. On the other hand, the determination of a partitioning membership of a point in a particular cluster takes into account the position of the point in the other clusters (relative to other clusters). Because of this, a partitioning can better capture the overlap that occurs between different clusters. Based on this discussion, we can see that in some situations where features are located in overlapping regions of clusters, the dual-partitioning approach based on its feature partitioning membership computations, can represent feature membership better than the existing FCCM model, which is based on its feature ranking membership computations. This is one advantage of the dual-partitioning approach over the FCCM model.

The second advantage of the dual-partitioning approach over the FCCM model is related to the way feature clusters are represented in general. Consider the matrix

$$D_2 = \begin{bmatrix} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & \begin{matrix} 1 \\ 1 \end{matrix} \end{bmatrix}$$

Figure 3.5: An Example D_2

D_2 in Figure 3.5, where rows and columns correspond to objects and features respectively. Naturally, we want to categorize the features (i.e. columns) into two clusters: $\{column_1, column_2, column_3\}$ and $\{column_4\}$. This categorization should ideally correspond to the feature membership matrix below, where rows and columns correspond to clusters and features respectively.

$$V = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

If we apply FCCM on D_2 , we instead get the following membership matrix:

$$V_{FCCM} = \begin{bmatrix} 0.33 & 0.33 & 0.33 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Even though the structure of the above membership matrix is similar to that of the ideal membership matrix V , the values of its elements do not represent the belongingness of a features to clusters naturally (i.e. 0.33 is small compared to 1). This is because these values are actually meant to represent the ranks of features. Furthermore, it should not be too difficult to picture, by deducing from the constraint in Eq. (2.12), that if the first feature or column cluster in D_2 were to contain 1000 features instead of 3 (i.e. $\{column_1, column_2, \dots, column_{1000}\}$ instead of $\{column_1, column_2, column_3\}$),

the membership value of each of this feature would be equal to 0.001 instead of 0.33. Hence, the more features contained in a cluster, the smaller the values of their ranking membership. In many practical cases, such as document-word co-clustering, the number of features (words) in every cluster can be in thousands. Consequently, the feature ranking membership would unrealistically take on very small values [103]. This reveals the unnatural representation that can occur in the FCCM model, which affects the interpretability of feature clusters. In contrast, applying any of the proposed dual-partitioning algorithm (i.e. FCR, HFCR, or IPFC) on D_2 , would result in the membership matrix below, which is consistent with the ideal membership matrix V we show above.

$$V_{\text{FCR}} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus, in many practical situations, the dual-partitioning approach can provide more natural representation of feature clusters than the FCCM model.

Other advantages of the dual-partitioning approach over the FCCM model can also come from the implementation perspective. In [124], we have explained that by resolving the problem of computational overflow in FCCM and the issue of the membership value range in Fuzzy CoDoK, the formulation of HFCR allows a more straightforward implementation than the two existing algorithms. This explanation can be extended for FCR and IPFC as well.

3.5.2 Disadvantages

Perhaps the most well-known weakness of a partitioning membership is its the sensitivity to outliers. We have explained this through an illustration in Figure 2.2 (in

Section 2.3.2 of Chapter 2) when discussing the motivation behind possibilistic clustering. Recall that given the situation in Figure 2.2, we would have the same partitioning membership values for the point P , which is an outlier, and the point Q , which is a non-outlier. Consequently, the dual-partitioning approach's performance can drop when outliers are present, because its partitioning memberships are unable to distinguish the outliers from the rest non-outliers. On the other hand, given the situation in Figure 2.2, a ranking membership would assign $v_{1P} < v_{1Q}$ and $v_{2P} < v_{2Q}$. This is because P , relative to other points, are further away from both C_1 and C_2 . Thus, in the existing FCCM model, outlier features would be assigned low ranking membership values, minimizing their impacts on the co-clustering process. This is one disadvantage of the dual-partitioning approach compared to the FCCM model.

The second problem with the dual-partitioning approach occurs when there are features that are equally important in more than one cluster. Consider an example of two clusters of articles: “soccer” and “basketball”. The term “match” should be equally important in both clusters, and therefore should have high membership values (close to 1) in there. Following the principle of a partitioning membership as explained earlier, the partitioning membership of this word however would be approximately 0.5 in both clusters. Therefore, the influence of the word has become less significant that it should have been ideally (i.e. drops from 1 to 0.5). On the other hand, the FCCM model allows the ranking membership values of such a word to be significant in the two clusters, if it has high rankings in both clusters.

We explain in Chapter 4 that by incorporating object and feature possibilistic memberships into our dual-partitioning approach model, the above two disadvantages can be mitigated.

3.6 Experimental Results

In this section, we present some experimental results to demonstrate the performance of the dual-partitioning approach for document categorization application. We used large document datasets which are taken from four benchmark document collections: *Classic3* in <ftp://ftp.cs.cornell.edu/pub/smart>, *20-Newsgroup* in <http://people.csail.mit.edu/jrennie/20Newsgroups>, *Reuters* in <http://www.daviddlewis.com/resources/testcollections/reuters21578>, and the large web page dataset (*WPD*) in [115]. We generated some subsets of these collections, as detailed in Table 3.5. We applied the three proposed dual-partitioning algorithms on the resulting eleven datasets listed in Table 3.5. For comparisons, experimental results of Fuzzy CoDoK, the variant of FCCM that can handle large datasets better [75], and HFCM, the cosine-distance variant of FCM, are also shown. We note that Fuzzy CoDoK here represents the existing FCCM model, and HFCM represents a standard fuzzy clustering model (as opposed to fuzzy co-clustering). Since we are dealing with document datasets, we use, in the following discussions, the terms “document” and “object” interchangeably. The same thing can be said for “word” and “feature”.

One important note is that, as stated earlier in this chapter, the main role of the dual-partitioning approach in this thesis is to provide the theoretical foundation for fuzzy-possibilistic co-clustering. Thus, the empirical discussions in this section are not extensive in nature. Our papers in [127, 124, 125] should provide more detailed empirical studies on the proposed dual-partitioning models.

3.6.1 Implementation Details

The Matrix Creation (MC-toolkit) [28] was utilized to pre-process the document collections. In this phase, we removed stopwords and words that occur too frequently and

Table 3.5: List of Document Datasets

Dataset	No. of Docs.	No. of Words	Clusters (No. of Docs)
<i>Classic3</i>	3893	2200	Medical (1033), Aerospace (1398), Inform. Retrieval (1460)
<i>Binary</i> (20-NG subset)	500	3377	Politics (250), Middle-east (250)
<i>Multi5</i> (20-NG subset)	500	2327	Computer Graphics (100), Motorcycle (100), Baseball (100), Space (100), Middle-east (100)
<i>Reuters3</i> (Reuters subset)	1076	2837	Trade (361), Crude (408), Money-fx (307)
<i>SS</i> (WPD subset)	2000	6337	Soccer (1000), Sport (1000)
<i>SM</i> (WPD subset)	2000	5450	Soccer (1000), Motorsport (1000)
<i>ABJ</i> (WPD subset)	3000	7688	Astronomy (1000), Biology (1000), Java (1000)
<i>CJB</i> (WPD subset)	2250	6781	Commercial Bank (1000), Java (750), Biology (500)
<i>CAB</i> (WPD subset)	2250	6980	Commercial Bank (1000), Astronomy (750), Biology (500)
<i>CB</i> (WPD subset)	2000	4791	Commercial Bank (1000), Building Societies (1000)
<i>WPD6s</i> (WPD subset)	600	2660	Commercial Bank (100), C/C++ (100), Astronomy (100), Biology (100), Soccer (100), Sport (100)

too rarely in the collections. In our experiments, d_{ij} is set to be equal to the normalized *tf-idf* value [111]. Information on the sizes of the document-word matrices, the number of clusters, and the topic of each cluster can be found in Table 3.5.

All the results presented in this paper are based on randomly initialized simulations. As all the algorithms involved in the experiments are not guaranteed to find global maximum, they were run several times. For this purpose, we adopted the strategy used in [86]. In this approach, three trials are performed for each test run. The trial that results in the most optimum objective function value is selected, except for HFCR (which has no explicit objective function) where we select the best trial according to the evaluation measures we use. The reported result is the average of ten of such test runs.

There are two fuzzifier parameters to be specified by users in FCR, HFCR, IPFC and Fuzzy CoDoK; i.e. T_u and T_v (by convention $T_u, T_v > 0$ [101]). In HFCM, there is one fuzzifier parameter m to be specified by users (by convention $m \geq 1$ [9]). In this study, we set these parameters empirically, that is, we ran a series of experiments with different parameter settings, and then chose the one that consistently resulted in good outcomes (i.e. the one with the highest 10-trial average accuracy). We used one important guideline as discussed in Section 2.3.1 of Chapter 2, that is, if we set the fuzzifier parameters too high, the resulting memberships become too fuzzy (i.e. close to $\frac{1}{C}$ for document and word partitioning memberships, or close to $\frac{1}{K}$ for word ranking membership), and if we set the fuzzifier parameters too low, the resulting memberships become too crisp (i.e. close to 0 or 1). These two extreme conditions usually do not give good performance: too fuzzy, the results are meaningless; too crisp, the membership representations become too restrictive. In our experiments, we usually starts by setting relatively high fuzzifier parameter values, and thus resulting in too fuzzy outcomes. We then gradually reduce the parameter values until we observe a significant

drop in the algorithm’s performance, indicating that the memberships have become too crisp. We then select the parameter settings that correspond to the most optimum performances. Based on our observations, all the five algorithms have comparable sensitivities to the fuzzifier parameter settings. We also observe that generally all these algorithms have certain ranges of fuzzifier parameter values, which correspond to their optimum performances. Within these ranges, we usually do not see that much variation in performances with respect to different fuzzifier parameter values. Furthermore, these ranges generally correspond to moderate levels of fuzziness in the resulting co-clusters (i.e. not too crisp and not too fuzzy). Table 3.6 tabulates these optimum parameter values. In our experiments, the number of clusters C was set according to the number of the ground-truth categories of documents given in each dataset, as indicated in Table 3.5. In all the experiments, the convergence-indicator parameter ϵ was set to 1×10^{-5} and the maximum limit of iteration τ_{max} was set to 200. We observe that all five algorithms consistently converged before the iteration reached this maximum limit of 200 iterations.

Table 3.6: Document Clustering Parameters Setting

Dataset	FCR		HFCR		IPFC		Fuzzy CoDoK		HFCM
	T_u	T_v	T_u	T_v	T_u	T_v	T_u	T_v	m
<i>Classic3</i>	1	1	0.001	0.001	2	2	0.00001	1.5	1.02
<i>Binary</i>	0.5	0.5	0.001	0.001	1	1	0.00001	1.5	1.05
<i>Multi5</i>	0.5	0.5	0.001	0.005	1	1	0.00001	1.5	1.05
<i>Reuters3</i>	1	1	0.01	0.01	1	1	0.00001	1.5	1.05
<i>SS</i>	1	1	0.001	0.001	5	5	0.001	1.5	1.02
<i>SM</i>	1	1	0.001	0.001	5	5	0.001	1.5	1.02
<i>ABJ</i>	2	3	0.001	0.001	2	2	0.001	1.5	1.02
<i>CJB</i>	2	3	0.001	0.001	2	2	0.00001	1.5	1.01
<i>CAB</i>	2	2	0.001	0.001	2	2	0.001	1.5	1.02
<i>CB</i>	2	2	0.001	0.003	5	7	0.001	1.5	1.03
<i>WPD6s</i>	1	0.7	0.001	0.001	1	1	0.001	1.5	1.02

For evaluation measure purpose, we applied a defuzzification process that assigned

every document to the cluster it has the highest membership to. After this assignment, the precision, recall, and purity measures were calculated. For every resulting document cluster c , the precision and recall are defined as follow [29]:

$$precision = \frac{|R_c|}{|R_c| + |S_c|}$$

$$recall = \frac{|R_c|}{|R_c| + |T_c|}$$

where R_c denotes the set of all documents correctly assigned to c , S_c denotes the set of all documents incorrectly assigned to c , and T_c denotes the set of all documents incorrectly not assigned to c . For every dataset, we recorded the average values of the precisions and recalls of all the clusters. To resolve any conflict in deciding which of the ground-truth categories a cluster represented, as much as possible we chose the most optimum assignment scheme (i.e. the one that resulted in the best average precisions and recalls). The purity is defined as follows [89]:

$$purity = \sum_{c=1}^C \frac{\max_{x \in X} |G_{cx}|}{|R_c| + |S_c|}$$

where X denotes the set of all ground-truth document categories, G_{cx} denotes the set of all documents in the ground-truth category x and assigned to cluster c , R_c and S_c are as defined above.

3.6.2 Results and Discussions

Table 3.7 shows the complete performance results of FCR, HFCR, IPFC, Fuzzy CoDoK, and HFCM. The standard deviations are shown in brackets. The main purpose of the dual-partitioning approach in our context is to build the dual-FCM-like foundation

for fuzzy-possibilistic co-clustering. It is therefore critical for the proposed models to function effectively in practical applications like document categorization. Comparisons with existing models like HFCM and Fuzzy CoDoK, which are known to work relatively well for document clustering can provide useful insight on this matter [95, 75]. Let us set aside the results of FCR from our discussions for now, because the algorithm overall do not give good document clustering performance.

Table 3.7: Document Clustering Performance

Dataset	FCR			HFCR			IPFC			FCodok			HFCM		
	Prec.	Rec.	Pur.	Prec.	Rec.	Pur.	Prec.	Rec.	Pur.	Prec.	Rec.	Pur.	Prec.	Rec.	Pur.
<i>Classic3</i>	98.6 (0)	98.3 (0)	98.6 (0)	98.6 (0.1)	98.5 (0)	98.6 (0)	98.5 (0)	98.4 (0)	98.4 (0)	98.6 (0)	98.33 (0)	98.46 (0)	98.9 (0)	98.8 (0)	98.8 (0)
<i>Binary</i>	68.65 (9.3)	62.2 (14.7)	62.2 (14.7)	88.73 (15.4)	88.72 (15.4)	88.72 (15.4)	89.21 (12.7)	89.1 (12.8)	89.1 (12.8)	86 (14.2)	85.96 (15)	85.96 (15)	87.65 (7.7)	85.8 (12.4)	85.8 (12.4)
<i>Multi5</i>	60.6 (7.6)	59.1 (11.8)	61.1 (9.4)	90.5 (6.2)	90 (6.6)	90.04 (6.6)	90.24 (8)	89.04 (10)	90.04 (7.9)	88.95 (11.3)	88.4 (11.1)	90.69 (11.1)	82.53 (10.1)	81.1 (9.7)	82.7 (8.4)
<i>Reuters3</i>	46.16 (11.1)	39.88 (3.1)	43.49 (2.4)	80.76 (4.3)	80.76 (4.7)	80.88 (4)	80.53 (2.6)	80.4 (3.1)	80.7 (2.7)	82.32 (10.2)	81.46 (11.1)	82.55 (9.9)	72.98 (6.8)	69.04 (4.7)	72.55 (3)
<i>SS</i>	56.61 (10.3)	52.9 (11.1)	52.9 (11.1)	88.94 (0)	88.9 (0.1)	88.9 (0.3)	88.19 (0)	88.15 (0)	88.15 (0)	82.01 (8.8)	81.44 (8.7)	81.44 (8.7)	83.74 (10.2)	83.32 (10.2)	83.32 (10.2)
<i>SM</i>	71.07 (7.9)	52.65 (1.1)	52.65 (1.2)	85.49 (3.4)	85.09 (3.3)	85.09 (3.3)	81.73 (3.2)	81.38 (3)	81.38 (3)	75.99 (5.2)	74.34 (3.7)	74.34 (3.7)	74.04 (7.1)	73.92 (7.2)	73.92 (7.2)
<i>ABJ</i>	51.6 (12.9)	34.7 (10.6)	36.3 (11.7)	85.18 (0)	84.8 (0)	84.8 (0.4)	86.05 (1.1)	85.74 (1.1)	85.74 (1.1)	83.61 (1.6)	78.26 (1.7)	78.26 (1.7)	59.5 (1.5)	58 (1.6)	61.3 (0.7)
<i>CJB</i>	81.82 (14.8)	43.9 (12.7)	51.51 (13)	88.36 (6.4)	88.91 (6.3)	89 (6.3)	90.58 (0.6)	91.47 (0.6)	91.34 (0.6)	89.37 (1.9)	87.5 (1)	88.82 (1.3)	83.46 (9.6)	85.24 (9.8)	85.74 (6.6)
<i>CAB</i>	79.31 (0)	58.08 (0)	69.16 (0)	83.2 (0.4)	83.4 (0.3)	85.7 (0.3)	82.4 (2.9)	82.35 (2.8)	84.88 (3)	79.95 (2.1)	78.83 (1.2)	80.35 (3)	67.18 (5.2)	67.38 (5.3)	74.01 (3.5)
<i>CB</i>	50 (0)	50 (0)	50 (0)	67.2 (2.8)	66.1 (2.7)	66.1 (2.7)	66.62 (3.3)	66.25 (3.3)	66.25 (3.3)	67.76 (5.7)	67.72 (5.6)	67.72 (5.6)	58.19 (1.4)	56.75 (0.6)	56.75 (0.6)
<i>WPD6s</i>	28.6 (6.7)	28.3 (5.9)	28.33 (1.3)	78.69 (4.0)	75.75 (3.9)	75.75 (3.9)	76.03 (6)	72.69 (6.9)	73.42 (6.9)	63.42 (5.5)	65.4 (4.6)	65.4 (4.6)	65.98 (6)	61.44 (5.4)	62.3 (5.1)

We can see from Table 3.7 that our fuzzy co-clustering algorithms HFCR and IPFC can consistently outperform the fuzzy clustering algorithm HFCM. This provides an affirmation that two of our dual-partitioning models can work properly for document clustering. That Fuzzy CoDoK can also generally perform better than HFCM in our experiments confirms that co-clustering, with suitable formulations, should give better performances than clustering in high-dimensional applications, as discussed in Section 2.2.7 of Chapter 2.

The results of HFCR, IPFC, and Fuzzy CoDoK in Table 3.7 gives another affirmation that dual-FCM can be successfully achieved in HFCR and IPFC. From the table, we can see that out of eleven datasets tested, HFCR and IPFC can give either better or comparable performances in nine datasets than Fuzzy CoDoK. There are however, two datasets (i.e. *Reuters3* and *CB*) where Fuzzy CoDoK gives better performances than the two dual-partitioning algorithms. In Section 3.5, we discuss analytically the advantages and disadvantages of the dual-partitioning and the FCCM model (as represented by Fuzzy CoDoK in our experiments) in different situations, including how the dual-partitioning may not perform well in the presence of outliers and how the FCCM model may not represent feature clusters naturally when in cases of overlapping co-clusters. To relate our analyses in Section 3.5 to the results we have here, there is a need for a deep analysis on several aspects of the document datasets we use, including the degrees of outliers and overlaps the datasets contained. Given the complexity of this task, and the fact that the dual-partitioning is not the end product that we are after in this thesis, we leave this more detailed study for future investigations.

Having shown that our dual-partitioning approach can be suitable for the document clustering task, we now discuss the comparisons of our three dual-partitioning models based on the empirical outcomes that we have. This discussion is essential for our expansion to fuzzy-possibilistic co-clustering. For clarity, the bar chart comparing the

Figure 3.6: Performance Comparison of FCR, HFCR, and IPFC

performances of FCR, HFCR, and IPFC, which corresponds to the results in Table 3.7, is shown in Figure 3.6. In this chart, the vertical axis gives the 10-test run average of *precision + recall + purity*. From the figure, on the one hand, we can see that IPFC and HFCR have relatively comparable performances, and can handle majority of the datasets very well. On the other hand, we observe that the performance of FCR is not as effective on all the datasets. This is likely due to FCR’s sensitivity to noise, as explained in Section 3.4.3. We note that, except for *Classic3*, the datasets that originated from *20Newsgroup*, *Reuters*, and *WPD* collections are noisy in nature [42, 138, 115]. Since we did not employ any feature selection during preprocessing, we might still have much noise contained in the datasets during co-clustering. This noise may negatively impact the performances of FCR. At this point we can conclude that IPFC and HFCR are the better models than FCR, as supported by our analysis and experimental results. From the results in Table 3.7 and Figure 3.6, between IPFC and HFCR, we cannot have a clear

conclusion as to which one is better from the performance perspective, since generally their performances are relatively comparable. Analytically however, IPFC is based on a firmer theoretical ground than HFCR, as the latter is heuristically formulated. On the other hand, HFCR's heuristic-based update rules are arguably simpler than those of IPFC (comparing Eqs. (3.12) and (3.13) with Eqs. (3.7) and (3.8)). Thus, the two algorithms do have their respective strong and weak points. In Chapter 4, we build a fuzzy-possibilistic co-clustering model based on the IPFC model. In Chapter 5, we formulate an alternative fuzzy-possibilistic co-clustering model, whose update rules are quite similar to the update rules of HFCR.

3.7 Conclusions

In this chapter, we have introduced three new models for fuzzy co-clustering based on the dual-partitioning approach, namely FCR, HFCR, and IPFC. These models address a limitation in the formulation of the existing FCCM model that creates a barrier to expand fuzzy co-clustering into fuzzy-possibilistic co-clustering. We have conducted extensive analytical study to justify the three proposed models. Among the three models proposed, we have examined that IPFC and HFCR are the better models than FCR. We have demonstrated the important points in our discussions through experiments on several large benchmark document datasets. The experimental results also confirm the effectiveness of the proposed methods. Finally, we have discussed some advantages and disadvantages of the dual-partitioning approach in general over the existing FCCM model. The disadvantages imply that further expansion of the dual-partitioning approach is needed. In the following two chapters, we introduce two such expansions in the framework of fuzzy-possibilistic co-clustering.

One possible future work is to study in more details how the advantages and dis-

advantages of the dual-partitioning approach and the FCCM model that we discuss in Section 3.5 can actually affect the fuzzy co-clustering performances in various real-world applications.

Chapter 4

The Dual Fuzzy-possibilistic Co-clustering

4.1 Overview

In this chapter, we present the details on the development of a new dual fuzzy-possibilistic co-clustering model based on IPFC, which is one of the dual-partitioning models we discuss in Chapter 3. The new algorithm, which we call the Dual Fuzzy-possibilistic Co-clustering (DFPC) algorithm, is formulated by incorporating into the IPFC's objective function, possibilistic object and feature memberships. A possibilistic membership captures a typicality measure between an object or a feature and a cluster [72], which is known to be robust to outliers. Thus, by incorporating possibilistic object and feature memberships into fuzzy co-clustering, we can get an improved performance in many real-world applications, where outliers are common. In addition, a possibilistic membership eliminates the notion of relative to other clusters in a partitioning membership, or relative to other features in a ranking membership, which, as discussed in Chapter 3, can cause misrepresentations of clusters in some situations [103]. This allows the

new algorithm to have better representations of object and feature memberships of co-clustering. As discussed in Chapter 2 however, a pure possibilistic clustering scheme suffers from the notorious coincident cluster problem [7]. One approach to tackle this weakness is by formulating a hybrid between fuzzy and possibilistic clustering as in the case of PFCM [103], a clustering algorithm which has been discussed in Chapter 2. In this context, the fuzzy clustering can provide the needed guidance to avoid the coincident cluster outcome [143, 103]. In DFPC, we bring the concept of hybridization between fuzzy and possibilistic approaches into the co-clustering domain. Because of this, DFPC is suitable for categorization of high-dimensional data. Furthermore, the simultaneous generation of fuzzy-possibilistic word clusters, which is absent in non co-clustering algorithms, enables DFPC to produce highly-interpretable co-clusters.

The remaining of this chapter is organized as follows. Section 4.2 discusses in details the motivation behind DFPC as our first fuzzy-possibilistic co-clustering algorithm. Section 4.3 explains the detailed formulation of DFPC. Our experimental results and discussions are given in Section 4.4. This chapter is concluded in Section 4.5.

4.2 The Motivation

We mention in Section 2.3.2 of Chapter 2 that a possibilistic membership measures the typicality of an object (or feature) to a cluster [72], that is, the membership depends solely on the similarity between the evaluated object and cluster (see the possibilistic memberships of PCM and PFCM in Eqs. (2.6) and (2.8) in Chapter 2). Unlike the partitioning and ranking memberships (we discuss them in Chapters 2 and 3), which capture the relative similarity to cluster (Section 3.5), a possibilistic membership captures the absolute similarity to cluster, as its value does not depend on the states of other clusters or other objects.

Because of its absolute nature, a possibilistic membership offers two important advantages. The first one is that it is robust to outliers, as discussed in Section 2.3.2 of Chapter 2. The second advantage is a better representation of clusters or co-clusters. We note that the misrepresentations in the ranking and partitioning memberships discussed in Section 3.5 of Chapter 3 are caused by the relative to other objects/features or relative to other clusters notions adopted by the two membership schemes. Since a possibilistic membership depends only on the absolute similarity between object/feature and cluster, these misrepresentations we discuss in 3.5 of Chapter 3 can be avoided by the possibilistic membership.

A pure possibilistic clustering algorithm however, suffers from the coincident-cluster problem [7], in which the outcome is in the form of several clusters (or all clusters in the extreme case) located exactly in the same location. The PFCM algorithm [103], as explained in Chapter 2, tries to combine FCM and PCM to remedy this problem. With the right setting of parameters, PFCM can simultaneously exhibit the PCM's robustness to outliers and good representation of clusters, and FCM's invulnerability to the coincident-cluster problem.

Inspired by this, we formulate a hybrid between possibilistic and fuzzy co-clustering in the proposed DFPC based on the dual-partitioning IPFC model introduced in Chapter 3. The motivation is to achieve robustness to object and feature outliers in co-clustering and an improved representation of co-clusters (possibilistic memberships), while avoiding the coincident-cluster problem (fuzzy partitioning memberships). Compared to PFCM, DFPC should be able to handle high-dimensional data better because of the implicit dimensionality reduction mechanism in the proposed algorithm through its simultaneous feature clustering. We use the IPFC model as the foundation of DFPC because, as we have studied in Chapter 3, its formulation has a strong theoretical basis and its performances are reasonably good based on our experiments on document

datasets.

4.3 The Proposed Model

In this section, the DFPC algorithm is explained in detail. The proof of convergence of the algorithm is also provided. Table 4.1 lists all the mathematical notations used in the following discussions.

Table 4.1: List of Mathematical Notations

Notation	Description
C, N, K	Numbers of (co-)clusters, objects (objects), and features (features)
u_{ci}, v_{cj}	Fuzzy partitioning object and feature memberships
x_{ci}, y_{cj}	Possibilistic object and feature memberships
d_{ij}	Relatedness measure between a object and a feature ($d_{ij} \geq 0$)
T_u, T_v, T_x, T_y	Co-clustering user-defined parameters
τ	Number of iterations
τ_{max}	Maximum number of iterations parameter
ϵ	Convergence indicator parameter

4.3.1 The DFPC Algorithm

The objective function of DFPC is given in Eq. (4.1). The goal is to maximize this function, subject to the constraints in Eqs. (2.2), (3.1), and (4.2).

$$\begin{aligned}
 J_{DFPC} = & 2 \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K (u_{ci} + x_{ci}) (v_{cj} + y_{cj}) d_{ij} - \\
 & \sum_{i=1}^N \left(\frac{\sum_{c=1}^C \sum_{j=1}^K (u_{ci} + x_{ci}) (v_{cj} + y_{cj})}{K} \sum_{j=1}^K d_{ij} \right) - \sum_{j=1}^K \left(\frac{\sum_{c=1}^C \sum_{i=1}^N (u_{ci} + x_{ci}) (v_{cj} + y_{cj})}{N} \sum_{i=1}^N d_{ij} \right) \\
 & - T_u \sum_{c=1}^C \sum_{i=1}^N u_{ci} \ln u_{ci} - T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj} \ln v_{cj} - T_y \sum_{c=1}^C \sum_{j=1}^K y_{cj}^2 - T_x \sum_{c=1}^C \sum_{i=1}^N x_{ci} \ln x_{ci}
 \end{aligned} \tag{4.1}$$

$$\sum_{c=1}^C \sum_{i=1}^N x_{ci} = N \tag{4.2}$$

From the objective function, we can capture that DFPC is an expansion of IPFC, an algorithm based on the dual-partitioning approach we discuss in Chapter 3. In DFPC, we have two object memberships u_{ci} and x_{ci} , and two feature memberships v_{cj} and y_{cj} . Before proceeding with the explanation, let us first derive, by applying the Lagrange multiplier method, the update rules for the DFPC optimization shown in Eqs. (4.3)-(4.6), where $s_{ci} = u_{ci} + x_{ci}$ and $r_{cj} = v_{cj} + y_{cj}$.

$$u_{ci} = \frac{\exp \left[\frac{1}{T_u} \left\{ 2 \sum_{j=1}^K r_{cj} d_{ij} - \frac{\sum_{j=1}^K r_{cj}}{K} \sum_{j=1}^K d_{ij} - \sum_{j=1}^K \left(\frac{r_{cj}}{N} \sum_{p=1}^N d_{pj} \right) \right\} \right]}{\sum_{e=1}^C \exp \left[\frac{1}{T_u} \left\{ 2 \sum_{j=1}^K r_{ej} d_{ij} - \frac{\sum_{j=1}^K r_{ej}}{K} \sum_{j=1}^K d_{ij} - \sum_{j=1}^K \left(\frac{r_{ej}}{N} \sum_{p=1}^N d_{pj} \right) \right\} \right]} \quad (4.3)$$

$$v_{cj} = \frac{\exp \left[\frac{1}{T_v} \left\{ 2 \sum_{i=1}^N s_{ci} d_{ij} - \frac{\sum_{i=1}^N s_{ci}}{N} \sum_{i=1}^N d_{ij} - \sum_{i=1}^N \left(\frac{s_{ci}}{K} \sum_{q=1}^K d_{iq} \right) \right\} \right]}{\sum_{e=1}^C \exp \left[\frac{1}{T_v} \left\{ 2 \sum_{i=1}^N s_{ei} d_{ij} - \frac{\sum_{i=1}^N s_{ei}}{N} \sum_{i=1}^N d_{ij} - \sum_{i=1}^N \left(\frac{s_{ei}}{K} \sum_{q=1}^K d_{iq} \right) \right\} \right]} \quad (4.4)$$

$$x_{ci} = \frac{N \exp \left[\frac{1}{T_x} \left\{ 2 \sum_{j=1}^K r_{cj} d_{ij} - \frac{\sum_{j=1}^K r_{cj}}{K} \sum_{j=1}^K d_{ij} - \sum_{j=1}^K \left(\frac{r_{cj}}{N} \sum_{p=1}^N d_{pj} \right) \right\} \right]}{\sum_{e=1}^C \sum_{f=1}^N \exp \left[\frac{1}{T_x} \left\{ 2 \sum_{j=1}^K r_{ej} d_{fj} - \frac{\sum_{j=1}^K r_{ej}}{K} \sum_{j=1}^K d_{fj} - \sum_{j=1}^K \left(\frac{r_{ej}}{N} \sum_{p=1}^N d_{pj} \right) \right\} \right]} \quad (4.5)$$

$$y_{cj} = \frac{1}{2T_y} \left\{ 2 \sum_{i=1}^N s_{ci} d_{ij} - \frac{\sum_{i=1}^N s_{ci}}{N} \sum_{i=1}^N d_{ij} - \sum_{i=1}^N \left(\frac{s_{ci}}{K} \sum_{q=1}^K d_{iq} \right) \right\} \quad (4.6)$$

From the corresponding membership constraints and the structures of the update equations in Eqs. (4.3) and (4.4) respectively, we can observe that the u_{ci} and v_{cj} of DFPC are fuzzy partitioning memberships of the same nature as the memberships in the dual-partitioning approach (or FCM). The y_{cj} equation in Eq. (4.6) shows that the membership depends solely on the similarity between the feature j and co-cluster c . This type of membership, as explained in Section 4.2, measures the typicality of features in co-clusters, and thus y_{cj} is a possibilistic membership. The x_{ci} in Eq. (4.5) is a possibilistic membership in a normalized form [80]. The numerator captures the similarity between a object i and a cluster c . The denominator has a common value across all $c = 1, \dots, C$ and $i = 1, \dots, N$. From these we know that the determining element in the x_{ci} equation (i.e. the numerator) depends only on the similarity of an object and a co-cluster. Hence x_{ci} is also a typicality measure. We adopt different styles of possibilistic memberships for x_{ci} and y_{cj} in order to have the values of all the four memberships in DFPC bounded on top and bottom. Note that based on the update equations, we have $0 \leq u_{ci}, v_{cj} \leq 1$ and $0 \leq x_{ci} \leq N$. Since $y_{cj} = f(u_{ci}, x_{ci})$, the membership value is bounded as well.

We now discuss the objective function of DFPC. The first three terms of J_{DFPC} in Eq. (4.1) function similarly as the first three terms of J_{IPFC} in Eq. (3.11) in Chapter 3. The first term is in a similar form to the original degree of aggregation in the FCCM model. Since $d_{ij} \geq 0$ for $i = 1, \dots, N$ and $j = 1, \dots, K$, it should not be too difficult to see that the maximization of this first term is biased towards co-clusters with larger $\sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K s_{ci} r_{cj}$ values, where $s_{ci} = u_{ci} + x_{ci}$ and $r_{cj} = v_{cj} + y_{cj}$. In other words

the optimization favors co-clusters with larger membership values. Consequently, this bias may shift the global maximum to a position corresponding to a poor-quality co-clustering [127]. Therefore in DFPC, to remove tackle this bias, we modify the degree of aggregation by incorporating the second and the third terms in Eq. (4.1). These second and third terms act as neutralizing factors to oppose the influence of the bias respectively at the document (i) and the word (j) levels. From their negative signs, it should be obvious that maximization of these terms will bring down the $\sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K s_{ci}r_{cj}$ value, providing a negative force to the bias. Furthermore, it can be shown that the first three terms in Eq. (4.1) is equal to the following with $s_{ci} = u_{ci} + x_{ci}$ and $r_{cj} = v_{cj} + y_{cj}$ as mentioned above:

$$\sum_{i=1}^N \sum_{j=1}^K \left[\left\{ 2 \sum_{c=1}^C s_{ci}r_{cj} - \frac{\sum_{c=1}^C \sum_{q=1}^K s_{ci}r_{cq}}{K} - \frac{\sum_{c=1}^C \sum_{p=1}^N s_{cp}r_{cj}}{N} \right\} d_{ij} \right]$$

and that the following always holds regardless of any individual membership value:

$$\sum_{i=1}^N \sum_{j=1}^K \left\{ 2 \sum_{c=1}^C s_{ci}r_{cj} - \frac{\sum_{c=1}^C \sum_{q=1}^K s_{ci}r_{cq}}{K} - \frac{\sum_{c=1}^C \sum_{p=1}^N s_{cp}r_{cj}}{N} \right\} = 0 = \text{constant}$$

The above shows that the DFPC's objective function possesses a bias-free property resembling the $\sum_{i=1}^N \sum_{j=1}^K \left\{ \sum_{c=1}^C u_{ci}v_{cj} \right\} = \text{constant}$ property in the FCCM model, which we discuss in Section 3.3 of Chapter 3.

The fourth and the fifth terms of Eq. (4.1) are the fuzzifiers of u_{ci} and v_{cj} respectively. They serve a similar purpose to the parameter m in FCM. The sixth term of Eq. (4.1) plays a similar role to the second term in the PCM's objective function (Eq. (2.5) in Chapter 2). The maximization of the first three terms of J_{DFPC} forces the values of $|y_{cj}|$ to be as large as possible. The maximization of the sixth term counters this

by pushing the membership values to zero. In a related fashion and considering the constraint in Eq. (4.2), the maximization of the first three terms of J_{DFPC} would make x_{ci} distributed as follow, $\vec{x} = (x_{11} = 0, x_{12} = 0, \dots, x_{ab} = N, \dots, x_{CN} = 0)$, if the cluster a and the object b is the most similar among other cluster-object pair's similarities. The maximization of the seventh term of Eq. (4.1) counters this by trying to achieve $x_{ci} = \frac{1}{C}$ for $c = 1, \dots, C$ and $i = 1, \dots, N$.

In the objective function J_{DFPC} in Eq. (4.1), we have the parameters T_u , T_v , T_x , and T_y to adjust the degrees of influence of the fuzzy and possibilistic elements of the algorithm, for both objects and features. If we increase the influence of the possibilistic memberships, the algorithm will be able to handle outliers better, but on the other hand it will also be very sensitive to initialization. If we increase the influence of the fuzzy memberships, the algorithm will be less sensitive to initialization, but also less robust to outliers.

All the four parameters should be assigned positive values. The following discussions can be deduced by observing the corresponding update memberships in Eqs.(4.3)-(4.6). In the case of T_u , a small value close to 0 will make u_{ci} very to either 0 or 1, while a large value will make u_{ci} very close to $\frac{1}{C}$ for all $c = 1, \dots, C$ and $i = 1, \dots, N$. Therefore, increasing the value of T_u reduces the influence of the fuzzy partitioning object membership u_{ci} , while decreasing the value of T_u enhances it. The parameter T_v has these same effects on the fuzzy partitioning feature membership v_{cj} . In the case of T_x , a small value close to 0 will make x_{ci} distributed approximately as follows, $\vec{x} = (x_{11} = 0, x_{12} = 0, \dots, x_{ab} = N, \dots, x_{CN} = 0)$, while a large value will make the values of x_{ci} close to $\frac{1}{C}$ for all $c = 1, \dots, C$ and $i = 1, \dots, N$. Therefore, increasing the value of T_x reduces the influence of the possibilistic object membership x_{ci} , while decreasing the value of T_x enhances it. In the case of T_y , a small value will increase the value of $|y_{cj}|$, while a large value will bring y_{cj} close to 0. Hence, increasing the value

of T_y reduces the influence of the possibilistic feature membership y_{cj} , while increasing the value of T_y enhances it.

Table 4.2 shows the pseudo-code of DFPC. The algorithm starts by randomly initializing u_{ci} and x_{ci} . It then starts iteratively updating all the four memberships until the objective function converges. DFPC has a time complexity of $O(CNK\tau)$, the same as PFCM and IPFC.

Table 4.2: DFPC Pseudo-code

DFPC ALGORITHM	
GIVEN: An $N \times K$ object-feature matrix (with d_{ij} as the element)	
PROCEDURE:	
1. Set parameters $T_u, T_v, T_x, T_y, \tau_{max}$, and ϵ ;	
2. Set $\tau = 0$;	
3. Randomly initialize u_{ci} and x_{ci} , $0 \leq u_{ci}, x_{ci} \leq 1$;	
4. REPEAT	
5. Update v_{cj} by Eq. (4.4);	
6. Update y_{cj} by Eq. (4.6);	
7. Update u_{ci} by Eq. (4.3);	
8. Update x_{ci} by Eq. (4.5);	
9. Update $\tau = \tau + 1$;	
10. UNTIL $\max_{c,i} u_{ci}(\tau) - u_{ci}(\tau - 1) \leq \epsilon$ or $\tau = \tau_{max}$;	

4.3.2 The Proof of Convergence

Theorems 7-11 discussed below leads to the proof of convergence of DFPC.

Theorem 7 *At every iteration τ , updating $u = (u_{11}, \dots, u_{CN})$ by Eq. (4.3) never decreases the objective function J_{DFPC} in Eq. (4.1).*

Proof. When updating u , we can consider the objective function J_{DFPC} as a function of u alone, denoted by $J_{DFPC}(u)$. This is because when updating u , we consider all the other memberships as constants with values computed from the previous iteration.

Theorem 7 can be proven by showing that the u^* (i.e u updated by Eq. (4.3)) is a local maximum of $J_{DFPC}(u)$ in the region observing the constraint in Eq. (2.2). From the fact that u^* is derived by applying the Lagrange multiplier method, we know that u^* has satisfied the first necessary condition and that it is a stationary point. But for it to be a local maximum, we need to show that the Hessian matrix $\nabla^2 J_{DFPC}(u^*)$ is negative definite. It can be shown that:

$$\begin{aligned} \nabla^2 J_{DFPC}(u) &= \begin{bmatrix} \frac{\partial J_{DFPC}(u)^2}{\partial u_{11}u_{11}} & \dots & \frac{\partial J_{DFPC}(u)^2}{\partial u_{11}u_{CN}} \\ \vdots & \ddots & \vdots \\ \frac{\partial J_{DFPC}(u)^2}{\partial u_{CN}u_{11}} & \dots & \frac{\partial J_{DFPC}(u)^2}{\partial u_{CN}u_{CN}} \end{bmatrix} \\ &= \begin{bmatrix} -\frac{T_u}{u_{11}} & & 0 \\ & \ddots & \\ 0 & & -\frac{T_u}{u_{CN}} \end{bmatrix} \end{aligned}$$

At u^* , we have $u_{ci} \geq 0$ for all $c = 1, \dots, C$ and $i = 1 \dots, N$ (from Eq. (4.3)). And we always assign a positive value to T_u . Therefore, the Hessian matrix $\nabla^2 J_{DFPC}(u^*)$ is negative definite. Since both the first necessary condition and the second sufficient condition ($\nabla^2 J_{DFPC}(u^*)$ is negative definite) are satisfied, u^* updated from Eq. (4.3) is indeed a local maximum of $J_{DFPC}(u)$ in the region observing the constraint in Eq. (2.2). For this reason, at every iteration, updating u by Eq. (4.3) never decreases the objective function J_{DFPC} . ■

Theorem 8 *At every iteration τ , updating $v = (v_{11}, \dots, v_{CK})$ by Eq. (4.4) never decreases the objective function J_{DFPC} in Eq. (4.1).*

Proof. Theorem 8 can be proven in a similar fashion as Theorem 7. ■

Theorem 9 *At every iteration τ , updating $x = (x_{11}, \dots, x_{CN})$ by Eq. (4.5) never decreases the objective function J_{DFPC} in Eq. (4.1).*

Proof. Theorem 9 can be proven in a similar fashion as Theorem 7. ■

Theorem 10 *At every iteration τ , updating $y = (y_{11}, \dots, y_{CK})$ by Eq. (4.6) never decreases the objective function J_{DFPC} in Eq. (4.1).*

Proof. Following a similar discussion as in Theorem 7, we can derive the following Hessian matrix:

$$\begin{aligned} \nabla^2 J_{DFPC}(y) &= \begin{bmatrix} \frac{\partial J_{DFPC}(y)^2}{\partial y_{11}y_{11}} & \dots & \frac{\partial J_{DFPC}(y)^2}{\partial y_{11}y_{CK}} \\ \vdots & \ddots & \vdots \\ \frac{\partial J_{DFPC}(y)^2}{\partial y_{CK}y_{11}} & \dots & \frac{\partial J_{DFPC}(y)^2}{\partial y_{CK}y_{CK}} \end{bmatrix} \\ &= \begin{bmatrix} -T_y & 0 \\ & \ddots \\ 0 & -T_y \end{bmatrix} \end{aligned}$$

Since we always assign a positive value to T_y , the Hessian matrix $\nabla^2 J_{DFPC}(y^*)$, where y^* is updated by Eq. (4.6), is negative definite. ■

Theorem 11 *J_{DFPC} in Eq. (4.1) is loosely bounded by a constant as follow:*

$$J_{DFPC} < \text{constant}, \text{ where constant} \neq \infty$$

Proof. To prove Theorem 8, we need to go through the terms in Eq. (4.1) one by one.

Let us consider the first three terms:

$$\begin{aligned}
& 2 \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K (u_{ci} + x_{ci}) (v_{cj} + y_{cj}) d_{ij} - \\
& \sum_{i=1}^N \left(\frac{\sum_{c=1}^C \sum_{j=1}^K (u_{ci} + x_{ci}) (v_{cj} + y_{cj})}{K} \sum_{j=1}^K d_{ij} \right) - \\
& \sum_{j=1}^K \left(\frac{\sum_{c=1}^C \sum_{i=1}^N (u_{ci} + x_{ci}) (v_{cj} + y_{cj})}{N} \sum_{i=1}^N d_{ij} \right)
\end{aligned}$$

If we can establish non-infinite range limits for u_{ci} , v_{cj} , x_{ci} , and y_{cj} , then the term above for certain has a non-infinite upper bound limit. From the update equations Eqs. (4.3)-(4.5), we know that $0 \leq u_{ci}, v_{cj} \leq 1$ and $0 \leq x_{ci} \leq N$. Based on these limits and from Eq. (4.6), it can be shown that the y_{cj} is loosely bounded: $\frac{1}{2T_y} \left\{ -\frac{\sum_{i=1}^N (1+N)}{N} \sum_{i=1}^N d_{ij} - \sum_{i=1}^N \left(\frac{(1+N)}{K} \sum_{q=1}^K d_{iq} \right) \right\} < y_{cj} < \frac{2 \sum_{i=1}^N (1+N) d_{ij}}{2T_y}$, i.e. $d_{ij} \geq 0$ for $i = 1, \dots, N$ and $j = 1, \dots, K$ as implied in Table 4.1. Since all the memberships are non-infinitely bounded, we have proven that the first three terms of Eq. (4.1) has a non-infinite upper bound limit.

Because of the constraints in Eqs. (2.2), (3.1), and (4.2), the following is true, which corresponds to the total fuzziness condition:

$$\begin{aligned}
& -T_u \sum_{c=1}^C \sum_{i=1}^N u_{ci} \ln u_{ci} - T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj} \ln v_{cj} - T_x \sum_{c=1}^C \sum_{i=1}^N x_{ci} \ln x_{ci} \leq -T_u N \ln \left(\frac{1}{C} \right) - \\
& T_v K \ln \left(\frac{1}{C} \right) - T_x N \ln \left(\frac{1}{C} \right)
\end{aligned}$$

Finally we also have $-T_y \sum_{c=1}^C \sum_{j=1}^K y_{cj}^2 \leq 0$. Since every term in Eq. (4.1) has a non-infinite upper bound limit, the Theorem 8 is true. ■

Corollary 3 *The DFPC algorithm in Table 4.2 converges to a local maximum of the*

optimization.

Proof. This corollary is a direct consequence of the five theorems above. Theorems 7 to 10 indicate that the procedure in Table 4.2 never decreases the DFPC objective function defined in Eq. (4.1). Theorem 11 states that there is a non-infinite limit to how much this objective function can be increased. So eventually the procedure should stop somewhere before or when it reaches this limit. ■

In summary, we have proven that under the updating rules of Eqs. (4.3)-(4.6) the algorithm DFPC converges to a local maximum of the objective function in Eq. (4.1) in the space constrained by Eqs. (2.2), (3.1), and (4.2).

4.4 Experimental Results

We divide this section into two subsections. In the first, we provide a numerical example to demonstrate the robustness of DFPC to outliers. In the second, we discuss our experiments on large benchmark document collections.

4.4.1 Robustness to Outliers

In this section, we demonstrate the robustness of DFPC to outliers through two numerical simulations. In the first we use a 2-D dataset to allow a relatively easier visualization and illustration of the results. In the second experiment, we use a high-dimensional dataset, which is the targeted application of the proposed algorithms.

2-D Dataset

Figure 4.1 shows the 2-D dataset used in this experiment, which contains 23 data points. From the figure, we can see two overlapping clusters, with one data point overlapped in

both clusters, and two outliers. Table 4.3 shows the coordinates and the ground truth labels of the 23 data points. Note that the point $i = 21$ corresponds to the one in the overlapping region between the two clusters, while the points $i = 22, 23$ correspond to the outliers.

Figure 4.1: 2-D Dataset with 23 Points

We applied DFPC to perform clustering on the dataset. For comparison, we also ran the same experiment using IPFC, HFCR, and Fuzzy CoDoK. For such a 2-D experiment, we are interested in the object membership values of the co-clustering algorithms, i.e. u_{ci} and x_{ci} of the DFPC, and the u_{ci} of the other three algorithms. Table 4.4 shows the detailed memberships generated by DFPC, IPFC, HFCR, and Fuzzy CoDoK. We can see that for all the four algorithms, the partitioning memberships u_{ci} allow us to capture the two overlapping clusters in the dataset. We note that we have high u_{C_1i} values and low u_{C_2i} values for $i = 1, \dots, 10$; and low u_{C_1i} values and high u_{C_2i} values for $i = 11, \dots, 20$. Furthermore, we also have $u_{C_121} = u_{C_221} \approx 0.5$ to indicate that the point $i = 21$ is positioned in the overlapping region between the two clusters. However, we can also observe that the outliers $i = 22, 23$ are assigned similar u_{ci} memberships as

Table 4.3: 2-D Dataset with 23 Points

d_{ij}	$j = 1$	$j = 2$	Cluster Label
$i = 1$	0.3	0.7	C_1
$i = 2$	0.25	0.65	C_1
$i = 3$	0.3	0.6	C_1
$i = 4$	0.4	0.6	C_1
$i = 5$	0.45	0.7	C_1
$i = 6$	0.35	0.7	C_1
$i = 7$	0.3	0.75	C_1
$i = 8$	0.2	0.75	C_1
$i = 9$	0.2	0.8	C_1
$i = 10$	0.35	0.8	C_1
$i = 11$	0.55	0.35	C_2
$i = 12$	0.55	0.5	C_2
$i = 13$	0.6	0.45	C_2
$i = 14$	0.6	0.35	C_2
$i = 15$	0.65	0.25	C_2
$i = 16$	0.7	0.35	C_2
$i = 17$	0.7	0.4	C_2
$i = 18$	0.7	0.5	C_2
$i = 19$	0.75	0.4	C_2
$i = 20$	0.75	0.3	C_2
$i = 21$	0.5	0.55	C_1 or C_2
$i = 22$	0.1	0.09	outlier
$i = 23$	0.09	0.1	outlier

the point $i = 21$, that is, $u_{C_1i} = u_{C_2i} \approx 0.5$ for $i = 22, 23$. This can mislead us to think that the two outliers are also positioned in the overlapping region of the two clusters. Figure 4.2 illustrates the data points with the cluster labels based on the resulting partitioning memberships u_{ci} of the four algorithms, namely DFPC, IPFC, HFCR, and Fuzzy CoDoK. We can see that in the figure, there is no distinction between the point $i = 21$ and the two outliers $i = 22, 23$ based on the u_{ci} to label the dataset.

Table 4.4: The Memberships of the 2-D Dataset with 23 Points

Point	DFPC's u_{ci}		DFPC's x_{ci}		IPFC's u_{ci}		HFRC's u_{ci}		Fuzzy CoDoK's u_{ci}	
	C_1	C_2	C_1	C_2	C_1	C_2	C_1	C_2	C_1	C_2
$i = 1$	0.73	0.27	0.77	0.29	0.67	0.33	0.70	0.30	0.74	0.26
$i = 2$	0.73	0.27	0.73	0.28	0.67	0.33	0.75	0.25	0.74	0.26
$i = 3$	0.67	0.33	0.64	0.31	0.63	0.37	0.70	0.30	0.68	0.32
$i = 4$	0.61	0.39	0.59	0.37	0.58	0.42	0.63	0.37	0.62	0.38
$i = 5$	0.64	0.36	0.68	0.38	0.61	0.39	0.67	0.33	0.65	0.35
$i = 6$	0.70	0.30	0.74	0.32	0.65	0.35	0.72	0.28	0.71	0.29
$i = 7$	0.75	0.25	0.85	0.28	0.70	0.30	0.78	0.22	0.77	0.23
$i = 8$	0.80	0.20	0.93	0.24	0.74	0.26	0.82	0.18	0.83	0.17
$i = 9$	0.82	0.18	1.02	0.23	0.75	0.25	0.84	0.16	0.86	0.14
$i = 10$	0.75	0.25	0.89	0.30	0.70	0.30	0.78	0.22	0.77	0.23
$i = 11$	0.35	0.65	0.32	0.58	0.39	0.61	0.37	0.63	0.38	0.62
$i = 12$	0.45	0.55	0.42	0.52	0.46	0.54	0.47	0.53	0.47	0.53
$i = 13$	0.39	0.61	0.37	0.59	0.41	0.59	0.40	0.60	0.41	0.59
$i = 14$	0.33	0.67	0.30	0.63	0.37	0.63	0.33	0.67	0.35	0.65
$i = 15$	0.25	0.75	0.24	0.74	0.30	0.70	0.25	0.75	0.26	0.74
$i = 16$	0.27	0.73	0.28	0.75	0.32	0.68	0.28	0.72	0.29	0.71
$i = 17$	0.30	0.70	0.31	0.72	0.34	0.66	0.30	0.70	0.32	0.68
$i = 18$	0.35	0.65	0.37	0.67	0.39	0.61	0.37	0.63	0.38	0.62
$i = 19$	0.27	0.73	0.29	0.79	0.32	0.68	0.28	0.72	0.29	0.71
$i = 20$	0.22	0.78	0.24	0.85	0.28	0.72	0.22	0.78	0.23	0.77
$i = 21$	0.51	0.49	0.49	0.46	0.51	0.49	0.53	0.47	0.53	0.47
$i = 22$	0.47	0.53	0.29	0.32	0.48	0.52	0.49	0.51	0.49	0.51
$i = 23$	0.49	0.51	0.30	0.31	0.49	0.51	0.51	0.49	0.51	0.49

Figure 4.2: 2-D dataset with 23 points labeled based on u_{ci}

In the case of DFPC however, the algorithm also gives us the possibilistic memberships x_{ci} , in addition to the partitioning memberships u_{ci} . Unlike the other three algorithms, the x_{ci} allows DFPC to detect the presence of outliers in a dataset. From Table 4.4, we can see that the x_{ci} values enable us to recognize the structure of the two overlapping clusters. Just like in the case of the u_{ci} , we have high x_{C_1i} values and low x_{C_2i} values for $i = 1, \dots, 10$; and low x_{C_1i} values and high x_{C_2i} values for $i = 11, \dots, 20$. We also have $x_{C_121} \approx x_{C_221}$ to indicate that the point $i = 21$ is located at the overlapping region of the two clusters. In addition, the x_{ci} values also enable us to recognize the two outliers $i = 22, 23$. We can see from the table that the values of x_{C_1i} and x_{C_2i} for $i = 22, 23$ are lower than those for $i = 1, \dots, 21$; that is, in the ranges of 0.29 to 0.32 in the former and 0.46 to 1.02 in the latter. From this, we can see that, based on the values of the possibilistic x_{ci} , it is possible to distinguish the outliers from the non-outliers in the dataset, as they should satisfy the condition $\max \{x_{C_1i}, x_{C_2i}\} < \theta$, where θ is equal to a relatively small figure. The distinction between outliers and non-outliers is not achievable in the case of u_{ci} , as we have seen earlier. The lower membership values

of the two outliers match well with our intuitions because outliers should have weak associations to any of the existing clusters, and hence their impact can be minimized in the clustering process. Figure 4.3 illustrates the data points with the cluster labels based on the resulting possibilistic memberships x_{ci} of DFPC. The figure shows how the two outliers are detected from their lower x_{ci} membership values to each of all the clusters C_1 and C_2 in the dataset.

Figure 4.3: 2-D dataset with 23 points labeled based on x_{ci}

From this 2-D experiment, we have demonstrated how the proposed fuzzy-possibilistic DFPC can be more robust in handling outliers than the existing fuzzy co-clustering counterparts, because of the availability of the possibilistic membership x_{ci} . Based on the results we just discussed, if we were to quantify the robustness measures of the four algorithms used in the experiment, we could say that DFPC manages to achieve 100% accuracy in robustness to outliers because it can detect both the two outliers present. On the other hand, IPFC, HFCR, and Fuzzy CoDoK fail to distinguish the two outliers from the non-outliers.

Table 4.5 shows the parameters used in this experiment for all the four algorithms.

We will discuss how to empirically set the parameters of DFPC in details in Section 4.4.2.

Table 4.5: Parameter Settings for the 2-D Experiments with 23 Points

Algorithm	T_u	T_v	T_x	T_y
DFPC	1	1	1	10
IPFC	1	0.5	-	-
HFCR	0.15	0.15	-	-
Fuzzy CoDoK	0.3	1	-	-

High-dimensional Dataset

We further extend our experiment to the case where we have a high-dimensional dataset. We use an artificial dataset D_{o1} , whose matrix representation is depicted in Figure 4.4. The rows of the matrix correspond to objects and the columns of the matrix correspond to features. The figure shows that the dataset contains two overlapping co-clusters represented by the submatrices C_1 and C_2 , and some outliers represented by the submatrix O . The sizes of the matrix (i.e. 60×80) and the submatrices are clearly indicated in the figure. For each of the co-clusters C_1 and C_2 , we assign each of its elements a real random value between 0.8 and 1, i.e. $0.8 \leq d_{ij} \leq 1$, indicating that the co-cluster's objects and features are closely related to each other. For the outliers in the submatrix O , we arbitrarily assign each of its elements a real random value between 0 and 1, i.e. $0 \leq d_{ij} \leq 1$. It should be reasonable to consider the elements of O as outliers due to their orthogonality to C_1 and C_2 , and the relatively smaller size of O as compared to the two co-clusters (i.e. 10×20 vs. 30×40). Other regions in D_{o1} , aside from C_1 , C_2 , and O , are assigned to 0.

We applied DFPC, IPFC, and Fuzzy CoDoK on D_{o1} (a similar experiment using HFCR will be presented in Chapter 5). We found that for this experiment, in DFPC, while different settings of the parameters T_u , T_v , T_x , and T_y change the numerical

Figure 4.4: Outlier-contaminated Dataset D_{o1}

outcomes of the memberships, they do not affect the conclusion that we got. We found the following parameters to give us a reasonable level of fuzziness (not too fuzzy nor too crisp) and a reasonable level of possibilistic or typicality (not too strong nor too weak): $T_u = 20$, $T_v = 20$, $T_x = 70$, and $T_y = 70$. Again, we will have more discussions on the empirical aspect of the DFPC's parameters in Section 4.4.2, including more details on how we systematically find the optimum parameter settings. For IPFC and Fuzzy CoDoK, we applied the following parameter settings, which were found using the procedure we explained in Section 3.6.1 of Chapter 3: $T_u = T_v = 5$ for IPFC, and $T_u = 0.25$, $T_v = 2$ for Fuzzy CoDoK. We set $\epsilon = 10^{-5}$ and $\tau_{max} = 200$ for this experiment.

For ease of evaluation, we divide the D_{o1} 's matrix into four row regions namely: $srow_1, \dots, srow_4$ and four column regions namely: $scol_1, \dots, scol_4$ as in Figure 4.4.

The row and column membership distributions should be uniform respectively on each of these row and column regions. For this reason, in Tables 4.6 and 4.7, rather than presenting every individual membership, we show the averages of the u_{ci} memberships of all the rows in each row regions, and the averages of the v_{cj} memberships of all the columns in each column regions.

Table 4.6: The Average Object Memberships of D_{o1}

Object	DFPC's u_{ci}			DFPC's x_{ci}			IPFC's u_{ci}			Fuzzy CoDoK's u_{ci}		
	C_1	C_2	$\sum_{c=1}^C u_{ci}$	C_1	C_2	$\sum_{c=1}^C x_{ci}$	C_1	C_2	$\sum_{c=1}^C u_{ci}$	C_1	C_2	$\sum_{c=1}^C u_{ci}$
$srow_1$	0.84	0.16	1	0.63	0.4	1.03	0.99	0.01	1	0.95	0.05	1
$srow_2$	0.5	0.5	1	0.58	0.58	1.16	0.5	0.5	1	0.5	0.5	1
$srow_3$	0.16	0.84	1	0.4	0.63	1.03	0.01	0.99	1	0.95	0.05	1
$srow_4$	0.5	0.5	1	0.36	0.36	0.72	0.5	0.5	1	0.5	0.5	1

Table 4.7: The Average Feature Memberships of D_{o1}

Feature	DFPC's v_{cj}			DFPC's y_{cj}			IPFC's v_{cj}			Fuzzy CoDoK's v_{cj}		
	C_1	C_2	$\sum_{c=1}^C v_{cj}$	C_1	C_2	$\sum_{c=1}^C y_{cj}$	C_1	C_2	$\sum_{c=1}^C v_{cj}$	C_1	C_2	$\sum_{c=1}^C v_{cj}$
$scol_1$	0.83	0.17	1	0.14	-0.1	0.04	0.9	0.1	1	0.002	0	0.002
$scol_2$	0.5	0.5	1	0.15	0.15	0.3	0.5	0.5	1	0.002	0.002	0.004
$scol_3$	0.17	0.83	1	-0.1	0.14	0.04	0.1	0.9	1	0	0.002	0.002
$scol_4$	0.5	0.5	1	-0.17	-0.17	-0.34	0.5	0.5	1	0	0	0

From Table 4.6, we can see that DFPC, Fuzzy CoDoK, and IPFC essentially share similar structures of object partitioning memberships u_{ci} . The three algorithms assign high u_{C_1i} values, and low u_{C_2i} values to objects in $srow_1$, indicating that they exclusively belong to C_1 . The reverse situation is true for objects in $srow_3$. For objects in $srow_2$ and $srow_4$, we have $u_{C_1i} = u_{C_2i} = \frac{1}{C} = 0.5$ to indicate their neutrality in the two co-clusters, which in the case of $srow_2$ is caused by the overlap between C_1 and C_2 , and hence the objects equally belong to the two co-clusters. In the case of object outliers in $srow_4$ however, their $\frac{1}{C}$ memberships are caused by the orthogonality, or equivalently their equally large distances to C_1 and C_2 . The u_{ci} values of objects in $srow_2$ and $srow_4$ reveal the limitation of Fuzzy CoDoK and IPFC, as the two algorithms treat outliers in $srow_4$ similarly as objects in $srow_2$, which are located in the overlapping region.

In the case of DFPC, an additional information provided by its object possibilistic membership x_{ci} enables the algorithm to reduce the influence of the outliers. As shown in Table 4.6, the value of $\sum_{c=1}^C x_{ci}$ is relatively smaller for objects in $srow_4$ as compared to the other objects in the matrix, including those in $srow_2$. This, as mentioned, is consistent with the common belief that outliers should have minimum contributions in any of the existing cluster, and hence their impact can be minimized in the co-clustering process. From this example, it is evident DFPC provides a more robust mechanism in dealing with object outliers.

In a similar manner, we can observe the robustness of DFPC to feature outliers from Table 4.7. Based on the feature partitioning memberships v_{cj} in DFPC and IPFC, we have $v_{C_1j} = v_{C_2j} = \frac{1}{C} = 0.5$ for the features in $scol_2$ (the features in the overlapping region) and the features in $scol_4$ (the feature outliers). Relying on these partitioning memberships alone, the algorithms make no distinction between the two cases. In the case of DFPC however, its feature possibilistic membership y_{cj} enables the algorithm to distinguish the feature outliers from the other features in the dataset. As shown in

Table 4.7, the value of $\sum_{c=1}^C y_{cj}$ is relatively smaller for features in $scol_4$ as compared to the other features in the matrix, including those in $scol_2$. The negative y_{cj} membership values assigned to the feature outliers indicate that these outliers bring negative contributions to the co-clustering process (i.e. to the maximization of the objective function). We can also notice from Table 4.7, that the feature ranking membership v_{cj} in Fuzzy CoDoK exhibits similar robustness to the feature possibilistic membership y_{cj} in DFPC, that is, the $\sum_{c=1}^C v_{cj} = 0$ for the feature outliers in $scol_4$. We have explained the robustness of a ranking membership to outliers in Section 3.5.2 of Chapter 3, which is one of the advantages of the existing FCCM model.

Based on this experiment, we can conclude that DFPC, due to its possibilistic object and feature memberships, can provide better robustness to outliers as compared to IPFC and Fuzzy CoDoK in a high-dimensional context. Again, based on the results we just discussed, if we were to quantify the robustness measures of the DFPC, IPFC, and Fuzzy CoDoK, we could say that DFPC manages to achieve 100% accuracy in robustness to outliers because it can detect all the outliers present, both from the object and the feature perspective. On the other hand, IPFC and Fuzzy CoDoK fail to detect any existence of the outliers in the dataset.

4.4.2 Document Clustering

In this section, we apply DFPC on the same large benchmark document datasets that we use in Chapter 3. The details of these datasets can be found in Table 3.5 of Chapter 3. For comparisons, experimental results of IPFC, and Fuzzy CoDoK [75] are also shown. First, we discuss some details on the implementations. These are then followed by the results and discussions.

Implementation Details

The same preprocessing detailed in Section 3.6.1 of Chapter 3 were applied in this document clustering experiment. Furthermore, as in Chapter 3, we also adopted the strategy used in [86] to cater for the random nature of the algorithms. In this strategy, three trials are performed for each test run, and the one that results in the most optimum objective function value is selected. The result we report here is the average of ten of such test runs.

To decide the optimum values of T_u , T_v , T_x , and T_y in DFPC, we adopted an empirical approach by running a number of simulations with different parameter settings and selecting the one that results in the optimum performance. We use the following guideline to arrive at the optimum parameters. First, we set the values of all the four parameters to be high in such a way that the all values of u_{ci} , v_{cj} , and x_{ci} are close to $\frac{1}{C}$, and the value of y_{cj} is close to 0. We have discussed the impact the parameter values on the corresponding memberships in Section 4.3. In this initial setting, no co-cluster is formed because the influences of both the fuzzy partitioning and the possibilistic elements of co-clustering are minimized, as explained in Section 4.3. We then gradually reduce the values of T_u and T_v to increase the influence of the fuzzy clustering (the T_x and T_y are fixed). Generally, after the values of T_u and T_v are below certain thresholds, depending on the natures of datasets, the clustering result drops significantly because the fuzzy partitioning memberships have become too crisp (i.e. too rigid representation of co-clusters). At this point, we stop reducing the T_u and T_v , and select the values that give the best performance so far. With the T_u and T_v fixed at their respective optimum values, we then focus on the possibilistic element of the co-clustering. The values of T_x and T_y are gradually decreased. After a certain point, when the values of these two parameters have become too low, the performance of DFPC should drop

significantly. This indicates that the influence of the possibilistic element of DFPC has become too strong, resulting in instability in the algorithm (i.e. the coincident cluster problem). At this point, we stop reducing the T_x and T_y , and select the values that give the best performance. Table 4.8 shows the list of optimum parameters we found for our experiments on all the datasets in Table 3.5.

Table 4.8: Document Clustering Parameters Setting

Dataset	DFPC			
	T_u	T_x	T_v	T_y
<i>Classic3</i>	0.5	100	3	100
<i>Binary</i>	5	50	0.1	1
<i>Multi5</i>	5	50	0.1	1
<i>Reuters3</i>	1	50	3	50
<i>SS</i>	5	50	5	100
<i>SM</i>	5	50	5	100
<i>ABJ</i>	2	50	2	100
<i>CJB</i>	2	50	2	100
<i>CAB</i>	2	50	2	100
<i>CB</i>	5	50	15	50
<i>WPD6s</i>	0.5	100	3	100

We observe from our experiments that DFPC generally performs the best in certain ranges of T_u , T_v , T_x , and T_y parameters that correspond to moderate level of fuzziness, i.e. the partitioning memberships u_{ci} and v_{cj} are not too fuzzy nor too crisp; and moderate level of typicality, i.e. the possibilistic memberships x_{ci} and y_{cj} are not too strong nor too weak. Within the boundaries of these optimum ranges, the algorithm's performance does not show a lot of variations with respect to any change in the parameter values. Formulating some mechanisms to allow the algorithm to automatically find its own optimum ranges of parameters can be considered as one potential future direction.

The other parameters, C , ϵ , and τ_{max} , were assigned by exactly the same values as in Section 3.6.1 of Chapter 3.

To evaluate the document clustering performance, each document was assigned to

the cluster with the largest document membership value. In DFPC, since we have two types of document membership, we calculate and use the $(u_{ci} + x_{ci})$ value for this purpose. Based on this assignment, the precision, recall, and purity measures were calculated [29, 89]. The formulas for these three measures can be found in Section 3.6.1 of Chapter 3.

Results and Discussions

Table 4.9 compares the performances of DFPC, IPFC, and Fuzzy CoDoK. The standard deviations are shown in brackets. The corresponding performance bar chart is shown in Figure 4.5. In this chart, the vertical axis gives the 10-test run average of *precision + recall + purity*. From the figure, we can observe that DFPC overall gives the best performances compared to the other algorithms. Compared to IPFC, DFPC gives mostly better performances, with a few exceptions (e.g. on *Classic3* and *CJB*) where the two algorithms give comparable performances. This finding indicates the usefulness of the possibilistic memberships in DFPC. The results in Figure 4.5 also suggest that the proposed fuzzy-possibilistic co-clustering model can outperform the existing FCCM model, as represented by Fuzzy CoDoK.

Table 4.9: Document Clustering Performance

Dataset	DFPC			IPFC			Fuzzy CoDoK		
	Prec.	Rec.	Pur.	Prec.	Rec.	Pur.	Prec.	Rec.	Pur.
<i>Classic3</i>	98.73 (0)	98.59 (0)	98.69 (0)	98.5 (0)	98.4 (0)	98.4 (0)	98.6 (0)	98.33 (0)	98.46 (0)
<i>Binary</i>	92.35 (11.5)	91.96 (11.5)	91.96 (11.6)	89.21 (12.7)	89.1 (12.8)	89.1 (12.8)	86 (14.2)	85.96 (15)	85.96 (15)
<i>Multi5</i>	90.97 (6.9)	90.87 (6.6)	91.76 (7.8)	90.24 (8)	89.04 (10)	90.04 (7.9)	88.95 (11.3)	88.4 (11.1)	90.69 (11.1)
<i>Reuters3</i>	84.52 (2.6)	84.29 (2.5)	83.73 (4)	80.53 (2.6)	80.4 (3.1)	80.7 (2.7)	82.32 (10.2)	81.46 (11.1)	82.55 (9.9)
<i>SS</i>	89.12 (0.1)	89.05 (0)	89.05 (0)	88.19 (0)	88.15 (0)	88.15 (0)	82.01 (8.8)	81.44 (8.7)	81.44 (8.7)
<i>SM</i>	82.76 (2.2)	82.45 (2.2)	82.45 (2.2)	81.73 (3.2)	81.38 (3)	81.38 (3)	75.99 (5.2)	74.34 (3.7)	74.34 (3.7)
<i>ABJ</i>	86.83 (0.3)	86.54 (1.1)	86.54 (0.4)	86.05 (1.1)	85.74 (1.1)	85.74 (1.1)	83.61 (1.6)	78.26 (1.7)	78.26 (1.7)
<i>CJB</i>	90.45 (1.1)	91.73 (1)	91.36 (1)	90.58 (0.6)	91.47 (0.6)	91.34 (0.6)	89.37 (1.9)	87.5 (1)	88.82 (1.3)
<i>CAB</i>	84.42 (0.4)	84.47 (0.6)	86.6 (0.5)	82.4 (2.9)	82.35 (2.8)	84.88 (3)	79.95 (2.1)	78.83 (1.2)	80.35 (3)
<i>CB</i>	68.32 (0.9)	68 (0.8)	68 (0.9)	66.62 (3.3)	66.25 (3.3)	66.25 (3.3)	67.76 (5.7)	67.72 (5.6)	67.72 (5.6)
<i>WPD6s</i>	80.99 (2.9)	73.27 (2.8)	73.27 (2.8)	76.03 (6)	72.69 (6.9)	73.42 (6.9)	63.42 (5.5)	65.4 (4.6)	65.4 (4.6)

Figure 4.5: Performance Comparison of DFPC, IPFC, and Fuzzy CoDoK

One area that should be improved due to the incorporation of the possibilistic memberships in DFPC is the interpretability of co-clusters. As an illustration, let us consider a few documents that score the lowest $\sum_{c=1}^C x_{ci}$ in the *Classic3* dataset. Figure 4.6 shows the full contents of the bottom four documents doc_1 - doc_4 . Compared to the more common lengthier documents in *Classic3*, e.g. doc_5 in Figure 4.7, documents doc_1 - doc_4 are merely short snippets, and they arguably exhibit more outlier-like quality than doc_5 . With possibilistic memberships, the distinction between these two groups of documents with different characteristics is made clear from the value of $\sum_{c=1}^C x_{ci}$, as shown in Table 4.10 (i.e. documents like doc_5 generally have large $\sum_{c=1}^C x_{ci}$ values). Contrast the possibilistic memberships in Table 4.10 with the corresponding fuzzy partitioning memberships shown in Table 4.11. Quite obviously, the partitioning memberships do not accommodate high interpretability as well as the possibilistic counterparts

Table 4.12 shows the top words with the highest membership values in the five

doc1:
jaundice in infancy most causes of jaundice in infancy
are discussed. the emphasis has been on those syndromes
solely or usually encountered in the neonate and infant.

doc2:
visual anosognosia in cortical blindness anton's
symptom. denial of blindness in a case of bilateral
hemianopia is presented to acquaint ophthalmologists with
this symptom which is well-known to neurologists
but is rarely seen by ophthalmologists.

doc3:
fine structure of subtilis phage sp-50. some structural
details and anomalous forms of subtilis phage sp-50
are described.

doc4:
Bradford's Law and the Keenan-Atherton Data Bradford's
methods are applied to the Keenan-Atherton data. The
results do not fit Bradford's Law.

Figure 4.6: Bottom 4 *Classic3* Documents with the lowest DPFC's $\sum_{c=1}^C x_{ci}$ values

clusters (or co-clusters) of *Multi5* generated by DFPC, IPFC, and Fuzzy CoDoK. Since DFPC has two types of word membership, i.e. v_{cj} and y_{cj} , the values of $v_{cj}+y_{cj}$ were used to decide the top words. It can be observed from the table that these words can generally describe the contexts of their corresponding document clusters, e.g. from words such as “nasa” and “comets”, one can deduce that space is the main context of document cluster C_1 . Based on the results in Table 4.12, we can argue one important benefit of DFPC over its fuzzy co-clustering algorithm counterparts, that is, DFPC gives two different perspectives of word clustering distributions, i.e. the possibilistic perspective (y_{cj}) and the fuzzy partitioning perspective (v_{cj}). Fuzzy co-clustering algorithms like IPFC or

Table 4.10: DFPC Possibilistic Memberships of *Classic3*'s doc_1 - doc_5

Document	C_1	C_2	C_3	$\sum_{c=1}^C x_{ci}$
doc_1	0.241	0.315	0.259	0.788
doc_2	0.274	0.308	0.233	0.815
doc_3	0.240	0.349	0.231	0.82
doc_4	0.260	0.261	0.300	0.821
doc_5	1.09	0.221	0.194	1.5

doc5:

on heat transfer over a sweat-cooled surface in laminar compressible flow with a pressure gradient. a simple expression is derived for the normal injection velocity distribution theoretically required to maintain a given uniform temperature along a porous surface in the laminar boundary-layer region of a compressible flow with a given velocity distribution outside of the boundary layer. this expression is valid for any given free-stream mach number but is based on a prandtl number of unity and on the assumption that the viscosity coefficient varies linearly with the temperature. by using the dorodnitsyn type of transformation, the variation of fluid properties even in the case of zero mach number is taken into account. this study is of particular practical interest in connection with the sweat-cooling of turbine blades and of airfoil surfaces in high speed flow. the method of analysis consists of applying the karman-pohlhausen method to both the momentum and energy boundary-layer equations and of using an additional heat balance equation, involving the coolant temperature. a closed-form approximate solution of the equations is then derived. numerical examples for flow in the immediate vicinity of a stagnation point and for a typical type of flow over a turbine blade are given.

Figure 4.7: Common *Classic3* DocumentsTable 4.11: DFPC Partitioning Memberships of *Classic3*'s doc_1 - doc_5

Document	C_1	C_2	C_3	$\sum_{c=1}^C u_{ci}$
doc_1	0	0.98	0.02	1
doc_2	0.01	0.99	0	1
doc_3	0	1	0	1
doc_4	0	0.01	0.99	1
doc_5	1	0	0	1

Fuzzy CoDoK, on the other hand, compute only one type of word membership. Hence, we can say that DFPC provides a more complete picture of how a word is set in a given dataset. Therefore the new algorithm is arguably more informative than its existing counterparts. In Chapter 6, we discuss how these informative word memberships of DFPC can be utilized to generate descriptive summaries of document clusters.

4.5 Conclusions

In this chapter, we have presented a new fuzzy-possibilistic co-clustering model, called DFPC, which captures simultaneously the advantages of fuzzy clustering, possibilistic clustering, and co-clustering. Its combined fuzzy-possibilistic formulation allows DFPC to provide rich representations of co-clusters, while exhibiting both robustness to outliers and stability with respect to the coincident problem. Its co-clustering formulation makes DFPC suitable for categorization of high-dimensional data, and at the same time able to capture the underlying theme of every resulting co-cluster through feature clustering. We have derived the new update rules and presented our analytical study on the new formulation. This analytical study, supported by our experimental results on large benchmark document datasets, suggest the superiority of the proposed fuzzy-possibilistic co-clustering over existing fuzzy co-clustering models. In Chapter 6, we discuss one additional potential of fuzzy-possibilistic co-clustering, that is, its ability to make document clusters more interpretable by feeding its resulting word clusters into an automatic document summarization engine.

We now discuss some possible future directions. One promising future work is to formulate a method to enable DFPC to automatically decide its own optimum parameter setting, including the optimum number of clusters. Second, DFPC assumes one-to-one correspondence between object and feature clusters, i.e. the numbers of object and fea-

Table 4.12: *Multi5*'s Top Words

Cluster (topic)	DFPC (v_{cj} , y_{cj})	IPFC (v_{cj})	Fuzzy CoDoK ($\frac{v_{cj}}{10^{-2}}$)
C_1 (space)	space (1.0, 6.29) nasa (1.0, 3.71) comet (1.0, 3.33) pat (1.0, 3.33) orbit (1.0, 2.78) station (1.0, 2.54) fred (1.0, 2.36) flight (1.0, 1.97)	space (0.999) nasa (0.997) comet (0.995) pat (0.994) orbit (0.982) fred (0.971) station (0.965) flight (0.934)	space (2.48) nasa (1.37) pat (1.28) comet (1.17) orbit (1.09) fred (0.95) station (0.94) nuclear (0.79)
C_2 (computer graphics)	graphics (1.0, 4.54) polygon (1.0, 2.92) algorithm (1.0, 2.75) code (1.0, 2.14) file (1.0, 2.09) vga (1.0, 1.86) image (1.0, 1.84) computer (1.0, 1.83)	graphics (0.999) polygon (0.991) algorithm (0.986) code (0.968) pov (0.948) vga (0.947) file (0.942) image (0.935)	graphics (1.99) polygon (1.21) algorithm (1.2) software (1.16) file (1.05) computer (1.0) code (1.0) pov (0.93)
C_3 (motorcycles)	bike (1.0, 4.46) dod (1.0, 2.89) ride (1.0, 2.47) jody (1.0, 2.4) riding (1.0, 2.38) levine (1.0, 1.92) drink (1.0, 1.76) back (1.0, 1.67)	bike (0.999) riding (0.98) dod (0.979) jody (0.977) ride (0.976) levine (0.95) drink (0.938) max (0.911)	bike (2.09) dod (1.42) riding (1.11) ride (1.04) jody (1.04) levine (0.84) bmw (0.8) max (0.76)
C_4 (baseball)	year (1.0, 3.86) players (1.0, 3.73) baseball (1.0, 3.32) season (1.0, 2.67) hitter (1.0, 2.6) win (1.0, 2.6) games (1.0, 2.54) bob (1.0, 2.45)	players (0.997) baseball (0.994) year (0.994) season (0.982) win (0.982) hitter (0.982) games (0.966) game (0.965)	year (1.34) players (1.31) baseball (1.12) game (1.06) season (0.94) win (0.91) hitter (0.9) bob (0.85)
C_5 (middle east)	israeli (1.0, 6.22) armenian (1.0, 5.97) turkish (1.0, 5.37) israel (1.0, 5.25) armenians (1.0, 4.52) people (1.0, 4.08) armenia (1.0, 3.99) arabs (1.0, 3.62)	israeli (1.0) armenian (1.0) turkish (1.0) israel (1.0) armenians (0.999) armenia (0.998) arabs (0.996) people (0.995)	israeli (1.89) armenian (1.56) israel (1.54) turkish (1.45) armenians (1.18) people (1.11) armenia (1.1) arabs (1.01)

ture clusters are the same. It is possible to reformulate a more flexible objective function where we can have many-to-many correspondence between object and feature clusters. Finally, in DFPC, we assign equal importance on both the fuzzy and the possibilistic components of the algorithm. It is possible to have a dynamic weighting mechanism so that we can assign different importance on the two components at different stages of optimization.

Chapter 5

The Possibilistic-fuzzy Co-clustering

A Simple Alternative Model for Fuzzy-possibilistic Co-clustering of Large Datasets

5.1 Overview

We report the details on the development of our second fuzzy-possibilistic co-clustering model, which we call Possibilistic-fuzzy Co-clustering (PFCC). Compared to the DFPC model discussed in Chapter 4, PFCC offers a simpler approach to perform fuzzy-possibilistic co-clustering on large datasets. We say on large datasets because the proposed model is formulated based on an assumption that clusters to be discovered contain significantly large numbers of objects. This is typically the case with large datasets, where clusters are considered significant if they contain many objects, while clusters containing relatively fewer objects are often regarded as outlier clusters [24, 13]. PFCC can uncover large size (i.e. containing many objects) clusters in a dataset, while filter out small size clusters by assigning low memberships to their members. If DFPC is strongly based on the dual-partitioning IPFC model discussed in Section 3.4.3 of Chap-

ter 3, PFCC, based on the structures of its update membership equations, as we show later in Section 5.2, is quite similar to the dual-partitioning HFCR model discussed in Section 3.4.2 of Chapter 3. We have discussed that the HFCR model results in relatively simpler update membership equations than the ones in IPFC. Consistent with this, the update membership equations of PFCC are relatively simpler than the DFPC counterparts. Hence, while the two algorithms have the same time complexity, given the same dataset, PFCC should require fewer computations than DFPC. As a fuzzy-possibilistic co-clustering model, PFCC offers similar benefits as DFPC, including robustness to outliers, interpretable representations of memberships, and stability with regard to the coincident cluster problem affecting pure possibilistic clustering algorithms [121].

Section 5.2 elaborates the detailed formulation of PFCC. In Section 5.3, analytical comparisons of PFCC and DFPC are provided. In Section 5.4, we discuss the experimental results. Section 5.5 concludes this chapter.

5.2 The PFCC Model

We first show the objective function of PFCC, and then proceed to derive the algorithm's update membership equations. The pseudo-code and the proof of convergence are discussed afterward. Table 5.1 shows the list of all the mathematical notations used in this chapter.

5.2.1 The Objective Function

The goal of PFCC is to maximize its objective function J_{PFCC} in Eq. (5.1), subject to the membership constraints in Eqs. (3.1) in Section 3.2 of Chapter 3 and (5.2) below.

Table 5.1: List of Mathematical Notations

Notation	Description
C, N, K	Numbers of (co-)clusters, objects (objects), and features (features)
x_{ci}	Possibilistic object membership
y_{cj}	Fuzzy ranking feature membership
v_{cj}	Fuzzy partitioning feature membership
d_{ij}	Relatedness measure between a object and a feature ($d_{ij} \geq 0$)
T_u, T_y, T_v	Co-clustering user-defined parameters
τ	Number of iterations
τ_{max}	Maximum number of iterations parameter
ϵ	Convergence indicator parameter

$$\begin{aligned}
J_{PFCC} = & \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K \left(\frac{x_{ci}}{\sum_{p=1}^N x_{cp}} \right) (v_{cj} + y_{cj}) d_{ij} - T_x \sum_{c=1}^C \sum_{i=1}^N (x_{ci})^2 \\
& - T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj} \ln v_{cj} - T_y \sum_{c=1}^C \sum_{j=1}^K y_{cj} \ln y_{cj}
\end{aligned} \tag{5.1}$$

$$\sum_{j=1}^N y_{cj} = 1 \text{ for } c = 1, 2, \dots, N \tag{5.2}$$

As indicated by the constraints, the PFCC objective function has three types of membership: object possibilistic membership x_{ci} , feature partitioning membership v_{cj} , and feature ranking membership y_{cj} . The first term of Eq. (5.1) is the modified version of the FCCM model's degree of aggregation which is going to be elaborated more shortly. The second term of Eq. (5.1) serves as a balancing mechanism for the object possibilistic membership. The maximization of the first term would force the value of x_{ci} to be as large as possible, while the maximization of the second term counter-balance this by forcing the value of x_{ci} to be as small as possible. The parameter T_x can be used to adjust how these two terms balance each other. A small value of T_x increases the value

of x_{ci} , while a large value of T_x does just the opposite. The third and the fourth terms of Eq. (5.1) are the fuzzifiers of v_{cj} and y_{cj} respectively. They control the levels of fuzziness of the two respective memberships. The two positive user-defined parameters T_v and T_y can be used to manually adjust the levels of fuzziness of the corresponding memberships. Based on the constraints in Eqs. (3.1) and (5.2), we have very high values of T_v and T_y respectively causing the $v_{cj} \approx \frac{1}{C}$ and $y_{cj} \approx \frac{1}{K}$ for all $c = 1, \dots, C$ and $j = 1, \dots, K$. On the other hand, very low values of T_v and T_y causes the two memberships to have values close to either 0 or 1.

We now try to explain the motivations behind the modified degree of aggregation. We first address why there are three types of membership. The object possibilistic membership is there to make PFCC less prone to the presence of outlier objects. The feature ranking membership is also included for the same reason, albeit this time is against the presence of feature outliers. In Section 3.5 of Chapter 2, we have discussed how a ranking membership is robust to outliers. We incorporate the feature partitioning membership for the same reason we have partitioning memberships in DFPC, i.e. to avoid the coincident cluster problem in a pure possibilistic approach, which can make an algorithm too sensitive to initialization. With a similar reasoning as in the possibilistic case, it can also be shown that the ranking membership alone also suffers from the coincident cluster problem [103]. Hence, we need a partitioning membership to avoid sensitivity to the initialization in PFCC. Unlike in DFPC however, where we have partitioning memberships for both objects and features, in PFCC, we rely solely on the feature partitioning membership. With one less variable to handle, PFCC can save some computational time as compared to DFPC. From our empirical study presented later in Section 5.4, we generally find that having partitioning membership for features only does not make PFCC more prone to the coincident cluster problem than DFPC. This is likely due to the nature of co-clustering, where clustering objects and features

are closely intertwined with each other [29], and therefore, stability on one side should cause stability on the other.

Next we explain the need to normalize x_{ci} (i.e. $\frac{x_{ci}}{\sum_{p=1}^N x_{cp}}$) in Eq. (5.1). Essentially the goal is to avoid the unwanted bias in the maximization of the degree of aggregation as explained in Section 3.3 of Chapter 3. The PFCC degree of aggregation can be represented as follows:

$$\begin{aligned} DoA_{PFCC} &= \sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K \left(\frac{x_{ci}}{\sum_{p=1}^N x_{cp}} \right) (y_{cj} + v_{cj}) d_{ij} = \check{\mathbf{M}} : \mathbf{D} \\ &= \begin{bmatrix} \check{m}_{11} & \cdots & \check{m}_{1K} \\ \vdots & \ddots & \vdots \\ \check{m}_{N1} & \cdots & \check{m}_{NK} \end{bmatrix} : \begin{bmatrix} d_{11} & \cdots & d_{1K} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{bmatrix} \end{aligned}$$

where $\check{m}_{ij} = \sum_{c=1}^C \left\{ \left(\frac{x_{ci}}{\sum_{p=1}^N x_{cp}} \right) (y_{cj} + v_{cj}) \right\}$ and $:$ denotes the component-wise inner product of two matrices. For a given matrix $\check{\mathbf{M}}$, we can calculate $\sum_{i=1}^N \sum_{j=1}^K \check{m}_{ij}$. Assuming that $\sum_{i=1}^N \sum_{j=1}^K \check{m}_{ij} \neq \text{constant}$, from the above expression, it can be seen that in the process of maximizing DoA_{PFCC} , a matrix $\check{\mathbf{M}}$ with a larger value of $\sum_{i=1}^N \sum_{j=1}^K \check{m}_{ij}$ tends to be more favored because such a matrix has the advantage to produce a larger DoA_{PFCC} value. Such is the case if x_{ci} is not normalized, and as explained in Chapter 3, this bias may cause a distortion in the optimization, which in turn can prevent the algorithm from producing optimum co-clustering. One can verify however that, with the x_{ci} normalized in PFCC, the value of $\sum_{i=1}^N \sum_{j=1}^K \check{m}_{ij}$ is always equal to a constant, i.e. $\sum_{i=1}^N \sum_{j=1}^K \check{m}_{ij} = K + C$. This prevents the disruptive bias towards larger $\sum_{i=1}^N \sum_{j=1}^K \check{m}_{ij}$ values in the optimization. We note that we have employed similar strategies in FCR and IPFC (in Sections 3.4.1 and 3.4.3 of Chapter 3), and DFPC (in

Section 4.3.1 Chapter 4) to tackle this bias problem.

Another important question to address is why PFCC incorporates feature ranking membership as opposed to feature possibilistic membership, just like in the case of DFPC. The reason for this is that if the feature ranking membership y_{cj} in J_{PFCC} were to be replaced with feature possibilistic membership (i.e. we can do this by not imposing the constraint in Eq. (5.2)), the property $\sum_{i=1}^N \sum_{j=1}^K \check{m}_{ij} = \text{constant}$ would no longer be preserved. This, as we have explained, can cause instability in the optimization. Despite its invulnerability to outliers, a ranking membership may suffer from representation problems in certain circumstances, as we discuss in Chapter 3. However, our discussions in Section 3.5 of Chapter 3 imply that such weaknesses can be minimized by combining ranking and partitioning memberships together in the formulation, that is, the disadvantages of a ranking membership are the advantages of a partitioning membership, and vice versa. This is precisely what we do in PFCC, that is, combining feature partitioning membership v_{cj} with feature ranking membership y_{cj} .

5.2.2 The Update Equations

To derive the PFCC update membership equations, we can apply the Lagrange multiplier method on the objective function J_{PFCC} in Eq. (5.1), taking into consideration the constraints in Eqs. (5.2) and (3.1). We now organized this section into two subsections. In the first one, the relatively simpler update equations for v_{cj} and y_{cj} are derived. In the second one, a special technique to derive the update equation of x_{ci} is presented.

v_{cj} and y_{cj}

By applying the Lagrange multiplier method on J_{PFCC} in Eq. (5.1), we can directly derive the following update feature membership equations:

$$v_{cj} = \frac{\exp\left(\frac{1}{T_v} \sum_{i=1}^N \frac{x_{ci}}{\sum_{p=1}^N x_{cp}} d_{ij}\right)}{\sum_{f=1}^C \exp\left(\frac{1}{T_v} \sum_{i=1}^N \frac{x_{fi}}{\sum_{p=1}^N x_{fp}} d_{ij}\right)} \quad (5.3)$$

$$y_{cj} = \frac{\exp\left(\frac{1}{T_y} \sum_{i=1}^N \frac{x_{ci}}{\sum_{p=1}^N x_{cp}} d_{ij}\right)}{\sum_{q=1}^K \exp\left(\frac{1}{T_y} \sum_{i=1}^N \frac{x_{ci}}{\sum_{p=1}^N x_{cp}} d_{iq}\right)} \quad (5.4)$$

By observing the equations structures, we can see that Eqs. (5.3) and (5.4) reflect partitioning and ranking memberships respectively. Furthermore, one can observe the striking similarity between the structures of the above update membership equations and the HFCR update membership equations discussed in Chapter 3.

An Incremental and Approximation Approach to Derive x_{ci}

If we try to derive x_{ci} directly by applying the Lagrange multiplier method on J_{PFCC} in Eq. (5.1), we will end up with a third degree polynomial problem. Instead of applying a direct mathematical approach to solve this problem, we devise a more practical alternative incremental approach that incorporates a simplified approximate version of the objective function. It is an incremental approach because the x_{ci} is updated one membership at a time. In each incremental update, the objective function in Eq. (5.1) is approximated by its relatively simpler version. The x_{ci} equation is then derived from the first necessary condition of this approximate objective function. There are two important points to be underlined: 1) the approximation is only valid under the Assumption 1 stated below, and 2) if Assumption 1 holds, we can prove that each incremental update of x_{ci} never decreases the objective function. The latter is important to ensure the algorithm's convergence (the proof of convergence of PFCC is detailed in Section 5.2.4).

Assumption 1 *At any point in the PFCC optimization process, we have $\sum_{p=1}^N x_{cp} \gg x_{ci}$ for all $c = 1, \dots, C$ and $i = 1, \dots, N$.*

Assumption 1 is an equivalent way of saying that there should be a significantly large number of objects in every cluster. This assumption is arguably reasonable for applications involving large datasets because in such a situation, clusters are generally made up of significantly large sets of objects. With regard to the presence of small outlier-like clusters in a dataset, since our algorithm utilizes the possibilistic clustering formulation, such outliers would be assigned small membership values, preventing the need to actually generate small size object clusters. In this way, the validity of Assumption 1 remains. In order to keep Assumption 1 maintained throughout the runtime of PFCC, it is important to ensure that the algorithm is not initialized with empty or small clusters. In practice we find that a random initialization is usually effective for this purpose.

To proceed with the derivation, let us express the objective function J_{PFCC} in Eq. (5.1) as a function of a single variable x_{ab} , as such is the case during a single incremental update of the possibilistic membership of an object b in a co-cluster a .

$$J_{PFCC}(x_{ab}) = \frac{\sum_{i=1}^N \left\{ x_{ai} \sum_{j=1}^K (v_{aj} + y_{aj}) d_{ij} \right\}}{\sum_{i=1}^N x_{ai}} - T_x (x_{ab})^2 + constant$$

$$\begin{aligned} \text{where } constant &= \sum_{\substack{c=1 \\ c \neq a}}^C \sum_{\substack{i=1 \\ i \neq b}}^N \sum_{j=1}^K \left\{ \left(\frac{x_{ci}}{\sum_{p=1}^N x_{cp}} \right) (v_{cj} + y_{cj}) d_{ij} \right\} - T_x \sum_{\substack{c=1 \\ c \neq a}}^C \sum_{\substack{i=1 \\ i \neq b}}^N (x_{ci})^2 \\ &\quad - T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj} \ln v_{cj} - T_y \sum_{c=1}^C \sum_{j=1}^K y_{cj} \ln y_{cj} \end{aligned}$$

Note that the terms y_{aj} and v_{aj} in the $J_{PFCC}(x_{ab})$ expression above are also considered as constants with values computed from the previous iteration. If Assumption 1 holds,

$J_{PFCC}(x_{ab})$ can be approximated by the function $\tilde{J}_{PFCC}(x_{ab})$ as follow:

$$\begin{aligned} J_{PFCC}(x_{ab}) &\approx \tilde{J}_{PFCC}(x_{ab}) \\ &= \frac{\sum_{i=1}^N \left\{ x_{ai} \sum_{j=1}^K (y_{aj} + v_{aj}) d_{ij} \right\}}{\sum_{\substack{i=1 \\ i \neq b}}^N x_{ai}} - T_x(x_{ab})^2 + constant \end{aligned}$$

From the first necessary condition of $\frac{\partial \tilde{J}_{PFCC}(x_{ab})}{\partial x_{ab}} = 0$, we can derive the update equation for x_{ab} .

$$x_{ab} = \frac{\sum_{j=1}^K (y_{aj} + v_{aj}) d_{bj}}{2T_x \sum_{\substack{i=1 \\ i \neq b}}^N x_{ai}} \quad (5.5)$$

Since $J_{PFCC}(x_{ab}) \approx \tilde{J}_{PFCC}(x_{ab})$, the stationery point of $\tilde{J}_{PFCC}(x_{ab})$ in Eq. (5.5) above should be a good estimate of the stationery point of $J_{PFCC}(x_{ab})$. In other words, if Assumption 1 holds, we can estimate the PFCC x_{ci} update equation by Eq. (5.5). The equation clearly reflects the characteristic of the possibilistic membership because it solely depends on the similarity between an object b and a co-cluster a (i.e. a typicality measure). This incremental and approximate approach is simpler and more practical to derive the x_{ci} update equation. Furthermore as long as Assumption 1 holds, it can be proven that each incremental update from Eq. (5.5) leads to the algorithm's convergence.

5.2.3 The Pseudo-code

Table 5.2 shows the pseudo-code of PFCC. Similar to DFPC, the new algorithm has a time complexity of $O(CNK\tau)$.

Table 5.2: PFCC Pseudo-code

PFCC ALGORITHM	
GIVEN: An $N \times K$ object-feature matrix (with $d_{ij} \geq 0$ as the element)	
PROCEDURE:	
1. Set parameters $T_x, T_v, T_y, C, \tau_{max}$, and ϵ ; 2. Set $\tau = 0$; 3. Randomly initialize x_{ci} , $0 \leq x_{ci} \leq 1$; 4. REPEAT 5. Update v_{cj} by Eq. (5.3); 6. Update y_{cj} by Eq. (5.4); 7. Update x_{ci} by Eq. (5.5); 8. Update $\tau = \tau + 1$; 9. UNTIL $\max_{c,i} x_{ci}(\tau) - x_{ci}(\tau - 1) \leq \epsilon$ or $\tau = \tau_{max}$;	

5.2.4 The Proof of Convergence

We shall prove that PFCC always converges under Assumption 1.

Theorem 12 *At every iteration τ , updating $v = \{v_{11}, \dots, v_{CK}\}$ from Eq. (5.3) never decreases the objective function J_{PFCC} in Eq. (5.1).*

Proof. When updating v , we can consider the objective function J_{PFCC} as a function of v alone, denoted by $J_{PFCC}(v)$. This is because when updating v , we can consider all the y_{cj} and x_{ci} terms in Eq. (5.1) as constants with values computed from the previous iteration. Theorem 12 can be proven by showing that the v^* (i.e. v updated by Eq. (5.3)) is a local maximum of $J_{PFCC}(v)$ in the region where the constraint in Eq. (3.1) is satisfied. From the fact that v^* is derived by applying the Lagrange multiplier method, we know that v^* has satisfied the first necessary condition and that it is a stationary point. But for it to be a local maximum, we need to show that the Hessian matrix $\nabla^2 J_{PFCC}(v^*)$ is negative definite. It can be shown that:

$$\nabla^2 J_{PFCC}(v) = \begin{bmatrix} \frac{\partial J_{PFCC}(v)^2}{\partial v_{11}v_{11}} & \dots & \frac{\partial J_{PFCC}(v)^2}{\partial v_{11}v_{CK}} \\ \vdots & \ddots & \vdots \\ \frac{\partial J_{PFCC}(v)^2}{\partial v_{CK}v_{11}} & \dots & \frac{\partial J_{PFCC}(v)^2}{\partial v_{CK}v_{CK}} \end{bmatrix} = \begin{bmatrix} -\frac{T_v}{v_{11}} & & 0 \\ & \ddots & \\ 0 & & -\frac{T_v}{v_{CK}} \end{bmatrix}$$

At v^* , we have $v_{cj} \geq 0$ for all $c = 1, \dots, C$ and $j = 1, \dots, K$ (from Eq. (5.3)). And we always assign a positive value to T_v . Therefore, the Hessian matrix $\nabla^2 J_{PFCC}(v^*)$ is negative definite. Since both the first necessary condition and the second sufficient condition ($\nabla^2 J_{PFCC}(v^*)$ is negative definite) are satisfied, v^* updated from Eq. (5.3) is indeed a local maximum of $J_{PFCC}(v)$ in the region where the constraint in Eq. (3.1) is satisfied. For this reason, at every iteration, updating v from Eq. (5.3) never decreases the objective function J_{PFCC} . ■

Theorem 13 *At every iteration τ , updating $y = \{y_{11}, \dots, y_{CK}\}$ from Eq. (5.4) never decreases the objective function J_{PFCC} in Eq. (5.1).*

Proof. Theorem 13 can be proven in a similar fashion as Theorem 12. ■

Theorem 14 *If Assumption 1 holds, each x_{ci} update from Eq. (5.5), for $c = 1, \dots, C$ and $i = 1, \dots, N$, never decreases the objective function J_{PFCC} in Eq. (5.1).*

Proof. It can be shown that $\frac{\partial \tilde{J}_{PFCC}(x_{ci})^2}{\partial^2 x_{ci}} = -2T_x$ ($\tilde{J}_{PFCC}(x_{ci})$ is defined in Section 5.2.2). Since T_x is always set to be positive, we have both the first necessary condition ($\frac{\partial \tilde{J}_{PFCC}(x_{ci})}{\partial x_{ci}} = 0$) and the second sufficient condition ($\frac{\partial \tilde{J}_{PFCC}(x_{ci})^2}{\partial^2 x_{ci}} < 0$) satisfied. Thus, the x_{ci}^* (i.e. x_{ci} updated from Eq. (5.5)) is a local maximum of $\tilde{J}_{PFCC}(x_{ci})$. If Assumption 1 holds, we have $J_{PFCC}(x_{ci}) \approx \tilde{J}_{PFCC}(x_{ci})$ and thus, the local maximum of $\tilde{J}_{PFCC}(x_{ci})$ should be a good estimate of the local maximum of $J_{PFCC}(x_{ci})$. From this we can conclude that if Assumption 1 holds, each x_{ci} update from Eq. (5.5) never decreases the objective function J_{PFCC} . ■

Theorem 15 *The following is true, with J_{PFCC} is as defined in Eq. (5.1).*

$$J_{PFCC} \leq 2C \sum_{i=1}^N \sum_{j=1}^K d_{ij} - T_v K \ln \left(\frac{1}{C} \right) - T_y C \ln \left(\frac{1}{K} \right)$$

Proof. The first term of J_{PFCC} can be expressed as follow:

$$\sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K \left(\frac{x_{ci}}{\sum_{p=1}^N x_{cp}} \right) (v_{cj} + y_{cj}) d_{ij} = \begin{bmatrix} \check{m}_{11} & \cdots & \check{m}_{1K} \\ \vdots & \ddots & \vdots \\ \check{m}_{N1} & \cdots & \check{m}_{NK} \end{bmatrix} : \begin{bmatrix} d_{11} & \cdots & d_{1K} \\ \vdots & \ddots & \vdots \\ d_{N1} & \cdots & d_{NK} \end{bmatrix}$$

where $\check{m}_{ij} = \sum_{c=1}^C \left\{ \left(\frac{x_{ci}}{\sum_{p=1}^N x_{cp}} \right) (v_{cj} + y_{cj}) \right\}$ and $:$ denotes the inner component-wise product of two matrices. Since $0 \leq \left(\frac{x_{ci}}{\sum_{p=1}^N x_{cp}} \right), v_{cj}, y_{cj} \leq 1$, we have $\check{m}_{ij} \leq 2C$, and consequently, the first term is loosely bounded:

$$\sum_{c=1}^C \sum_{i=1}^N \sum_{j=1}^K \left(\frac{x_{ci}}{\sum_{p=1}^N x_{cp}} \right) (v_{cj} + y_{cj}) d_{ij} \leq 2C \sum_{i=1}^N \sum_{j=1}^K d_{ij}$$

The second term of J_{PFCC} has an obvious maximum value of zero because x_{ci} is always positive (from Eq. (5.5)). The third and the fourth terms of J_{PFCC} are fuzzy entropies. Given the constraints, they are maximized when $v_{cj} = \frac{1}{C}$ and $y_{cj} = \frac{1}{K}$ for $c = 1, \dots, C$ and $j = 1, \dots, K$ respectively, which corresponds to the maximum entropy condition [22]. This translates into the following:

$$-T_v \sum_{c=1}^C \sum_{j=1}^K v_{cj} \ln v_{cj} - T_y \sum_{c=1}^C \sum_{j=1}^K y_{cj} \ln y_{cj} \leq -T_v K \ln \left(\frac{1}{C} \right) - T_y C \ln \left(\frac{1}{K} \right)$$

Putting everything together gives us the proof of Theorem 15. ■

We can now state the convergence of PFCC.

Corollary 4 *If Assumption 1 holds, the PFCC algorithm converges to a local maximum of the optimization, with the updates given in Eqs. (5.3), (5.4), and (5.5).*

Proof. This corollary is a direct consequence of the above four theorems. If Assumption 1 holds, Theorems 12 to 14 indicate that the procedure in Table 5.2 never decreases the PFCC objective function. Theorem 15 states that there is a limit to how much this objective function can be increased. So eventually the procedure should stop somewhere before or when it reaches this limit. ■

In summary, we have proven that, if Assumption 1 holds, under the updating rules of Eqs. (5.3)-(5.5) the algorithm PFCC converges to an approximate local maximum of the objective function in Eq. (5.1) in the space constrained by Eqs. (3.1) and (5.2).

5.3 PFCC and DFPC: An Analytical Comparison

In this section, we discuss the advantages and disadvantages of PFCC over DFPC from an analytical perspective. In the beginning of this chapter, we mentioned that PFCC provides a relatively simpler approach to fuzzy-possibilistic co-clustering as compared to DFPC. We now provide the justification for this by showing that PFCC requires fewer computations than DFPC. Firstly, as discussed in Section 5.2, the formulation of PFCC involves three different types of membership, while the formulation of DFPC involves

four different types of membership (i.e. the former does not include object partitioning membership in the objective function, while the latter does). Therefore compared to DFPC, PFCC has one less variable to compute. Secondly, even if we take the object partitioning membership out of consideration, we can still show that the computations of the other three types of membership are more complex in DFPC than in PFCC. Tables 5.3-5.5 contrast the procedures involved in updating the three memberships of PFCC and DFPC (excluding the object partitioning membership in DFPC) in one iteration. Therefore, we can conclude that PFCC involves fewer computations than DFPC, and thus, offers faster fuzzy-possibilistic co-clustering in real time.

Table 5.3: Real-time Computations Comparison: PFCC's v_{cj} vs. DFPC's v_{cj}

PFCC Eq. (5.3)	Runtime	DFPC Eq. (4.4)	Runtime
1. Calculate $numerator_{cj}$: a) $\sum_{i=1}^N x_{ci}$ b) $\frac{\sum_{i=1}^N x_{ci} d_{ij}}{\sum_{i=1}^N x_{ci}}$	$O(CN)$ $O(CNK)$	1. Calculate $numerator_{cj}$: a) $\sum_{i=1}^N s_{ci}$ b) $2 \sum_{i=1}^N \left(s_{ci} - \frac{\sum_{p=1}^N s_{cp}}{N} \right) d_{ij}$ c) $\sum_{i=1}^N \left(\frac{s_{ci}}{K} \sum_{q=1}^K d_{iq} \right)$	$O(CN)$ $O(CNK)$ $O(CNK)$
2. Calculate denominator: a) $\sum_{c=1}^C numerator_{cj}$	$O(CK)$	2. Calculate denominator: a) $\sum_{c=1}^C numerator_{cj}$	$O(CK)$
Total: $O(CN) + O(CNK) + O(CK)$		Total: $O(CN) + 2O(CNK) + O(CK)$	

On the downside however, unlike DFPC, which is applicable for any kind of situation, PFCC is applicable strictly when Assumption 1 is acceptable. This assumption, we have mentioned, is likely to be reasonable when dealing with large datasets containing large size clusters, e.g. large document collections. In conclusion, PFCC, as compared to DFPC, can provide a simpler and more efficient fuzzy-possibilistic co-clustering in its

Table 5.4: Real-time Computations Comparison: PFCC's y_{cj} vs. DFPC's y_{cj}

PFCC Eq. (5.4)	Runtime	DFPC Eq. (4.6)	Runtime
1. Calculate $numerator_{cj}$: a) Calculate: $\sum_{i=1}^N x_{ci}$ b) $\frac{\sum_{i=1}^N x_{ci}d_{ij}}{\sum_{i=1}^N x_{ci}}$	$O(CN)$ $O(CNK)$	1. Calculate: a) $\sum_{i=1}^N s_{ci}$ b) $2 \sum_{i=1}^N \left(s_{ci} - \frac{\sum_{p=1}^N s_{cp}}{N} \right) d_{ij}$ c) $\sum_{i=1}^N \left(\frac{s_{ci}}{K} \sum_{q=1}^K d_{iq} \right)$	$O(CN)$ $O(CNK)$ $O(CNK)$
2. Calculate denominator: a) $\sum_{j=1}^K numerator_{cj}$	$O(CK)$		
Total: $O(CN) + O(CNK) + O(CK)$		Total: $O(CN) + 2O(CNK)$	

limited domain of application.

5.4 Experimental Results

In this section we present the empirical study of PFCC. We first provide a numerical example to demonstrate the robustness of PFCC to outliers, and then discuss the algorithm's performance when being applied on large document datasets.

5.4.1 Robustness to Outliers

In this section, we conduct a similar simulation as in Section 4.4.1 of Chapter 4 to demonstrate the robustness of PFCC to outliers. We discuss the algorithm's performance when applied on two datasets. The first dataset is a 2-D dataset, which allows an easier visualization of the experiment. The second dataset is a high-dimensional dataset, which is the target application of the algorithm.

Table 5.5: Real-time Computations Comparison: PFCC's x_{ci} vs. DFPC's x_{ci}

PFCC Eq. (5.5)	Runtime	DFPC Eq. (4.5)	Runtime
1. Calculate: a) $\sum_{p=1, p \neq i}^N x_{cp}$	$O(CN^2)$	1. Calculate $numerator_{ci}$: a) $\sum_{j=1}^K r_{cj}$	$O(CK)$
b) $\sum_{j=1}^K (y_{cj} + v_{cj}) d_{ij}$	$O(CNK)$	b) $2 \sum_{j=1}^K \left(r_{cj} - \frac{\sum_{q=1}^K r_{cq}}{K} \right) d_{ij}$	$O(CNK)$
		c) $\sum_{j=1}^K \left(\frac{r_{cj}}{N} \sum_{p=1}^N d_{pj} \right)$	$O(CNK)$
		2. Calculate denominator: a) $\sum_{c=1}^C \sum_{i=1}^N numerator_{ci}$	$O(CN)$
Total: $O(CN^2) + O(CNK)$		Total: $O(CK) + 2O(CNK) + O(CN)$	

2-D Dataset

The experiment conducted in this section is similar to the one in Section 4.4.1. However, we need to take into account the Assumption 1 underlying the formulation of PFCC, namely the number of objects in any cluster should be significantly large (we have more discussions on the validity of Assumption 1 in Section 5.4.2 later). For this reason, we use the 2-D dataset shown in Figure 5.1, in which we have two overlapping clusters each containing a relatively large number of data points. To be precise, we have 267 points belong exclusively to each of the two clusters, and 66 points in the overlapping region between the two clusters. In addition, we also have 2 outlier points as shown in the figure. So in total we have 602 data points in the dataset. As we are dealing with a large number of points, it is easier to discuss in terms of the regions the points are located in instead of every individual points. Hence, as illustrated in the figure, we denote with region A all the points that exclusively belong to one of the clusters, region B all the points that exclusively belong to the other cluster, region C all the points

that are positioned in the overlapping region between the two clusters, and region D the outliers. We note that the points in each region should have similar fuzzy (and possibilistic) membership distributions to one another.

Figure 5.1: 2-D Dataset with 602 Points

We applied PFCC on the dataset. For comparison, we also ran the same experiment on IPFC, HFCR, and Fuzzy CoDoK. For such a 2-D experiment, we are interested in the object membership values of the co-clustering algorithms, i.e. the x_{ci} of PFCC, and the u_{ci} of IPFC, HFCR, and Fuzzy CoDoK. Table 5.6 shows the related membership values generated by PFCC, IPFC, HFCR, and Fuzzy CoDoK. Since, as mentioned, the points in each region should have similar membership distributions to one another, we show the averages of the memberships in each region instead of the membership of every individual of the 602 points. From these memberships, we can observe that all the object memberships of the four algorithms, either the possibilistic membership x_{ci} or the partitioning membership u_{ci} , can capture the structure of the two overlapping clusters. We note that in the case of PFCC, we have high x_{C_1i} values and low x_{C_2i} values for the points in the region A , indicating that they exclusively belong to the one

of the clusters; and low x_{C_1i} values and high x_{C_2i} values for the points in the region B , indicating that they exclusively belong to the other cluster. We also have $x_{C_1i} \approx x_{C_2i}$ for the points in the region C , representing that these points are equally shared between the two clusters. Up to this point, the u_{ci} of all the other three algorithms are also able to correctly capture the underlying structure. However, when it comes to the outliers in the region D , there is a difference in interpretation of data point i between PFCC and the others such as IPFC/HFCR/Fuzzy CoDoK, where x_{ci} and u_{ci} are used respectively. From Table 5.6, we can see how the points in the region D have lower memberships x_{C_1i} and x_{C_2i} than those in the other regions. On the other hand, we have the u_{C_1i} and u_{C_2i} for the points in the region D more or less equal to the ones for the points in the overlapping region C . Hence, in this experiment, only PFCC can identify the presence of outliers. By assigning lower memberships to outliers, PFCC can minimize the impact of outliers in the clustering process. Figures 5.2 and 5.3 depict the clusters captured by the partitioning memberships u_{ci} of IPFC, HFCR, and Fuzzy CoDoK; and the possibilistic memberships x_{ci} of PFCC respectively. Note that in the latter, the outliers are properly distinguished from the rests.

Table 5.6: The Average Memberships of the 2-D Dataset with 602 Points

Region	PFCC's x_{ci}		IPFC's u_{ci}		HFCR's u_{ci}		Fuzzy CoDoK's u_{ci}	
	C_1	C_2	C_1	C_2	C_1	C_2	C_1	C_2
A	0.51	0.35	0.35	0.65	0.68	0.32	0.38	0.62
B	0.30	0.47	0.65	0.35	0.32	0.68	0.62	0.38
C	0.42	0.41	0.50	0.50	0.51	0.49	0.51	0.49
D	0.08	0.08	0.50	0.50	0.50	0.50	0.50	0.50

From this 2-D experiment, we have demonstrated how the fuzzy-possibilistic co-clustering PFCC, through its possibilistic membership x_{ci} , can be more robust to outliers than the existing fuzzy co-clustering counterparts. Based on the results, if we were to

Figure 5.2: 2-D dataset with 602 points labeled based on u_{ci}

Figure 5.3: 2-D dataset with 602 points labeled based on x_{ci}

quantify the robustness measures of the four algorithms used in the experiment, we could say that PFCC manages to achieve 100% accuracy in robustness to outliers because it can detect both the two outliers present. On the other hand, IPFC, HFCR, and Fuzzy CoDoK fail to distinguish the outliers from the non-outliers.

Table 5.7 shows the parameter settings of all the four algorithms. We will discuss how to empirically set the parameters of PFCC in details in Section 5.4.2.

Table 5.7: Parameter Settings for the 2-D Experiments with 602 Points

Algorithm	T_u	T_v	T_x	T_r
PFCC	-	0.03	0.005	0.03
IPFC	0.5	0.5	-	-
HFCR	0.13	0.13	-	-
Fuzzy CoDoK	0.5	1	-	-

High-dimensional Dataset

We now extend our experiment to a high-dimensional dataset. We use an artificial dataset D_{o2} , whose size is 600×800 , which is ten times the size of D_{o1} in Chapter 4. We increase the size of the dataset in order to satisfy Assumption 1 for PFCC to function properly (more on this validity of Assumption 1 in Section 5.4.2). The matrix representation of D_{o2} is depicted in Figure 5.4. The rows of the matrix correspond to objects and the columns of the matrix correspond to features. The sizes of the matrix and the submatrices are clearly indicated in Figure 5.4. The figure shows two overlapping co-clusters represented by the submatrices C_1 and C_2 in the dataset, and some outliers represented by the submatrix O . The sizes of these submatrices are indicated in the figure. For each of the co-clusters C_1 and C_2 , we assign each of its elements a real random value between 0.8 and 1, i.e. $0.8 \leq d_{ij} \leq 1$, indicating that the co-cluster's objects and features are closely related to each others. For the outliers in the submatrix O , we arbitrarily assign each of its elements a real random value between

0 and 1, i.e. $0 \leq d_{ij} \leq 1$. We can consider the elements of O as outliers because of their orthogonality to C_1 and C_2 , and the relatively smaller size of O as compared to the two co-clusters (i.e. 100×200 vs. 300×400). Other regions in D_{o2} , aside from C_1 , C_2 , and O , are assigned to 0.

Figure 5.4: Outlier-contaminated Dataset D_{o2}

We applied PFCC, HFCR, IPFC, and Fuzzy CoDoK on D_{o2} . Similar to DFPC, we found that PFCC was not sensitive to parameter settings for this particular experiment. We found the following parameters to give us a reasonable level of fuzziness (not too fuzzy nor too crisp) and a reasonable level of possibilistic or typicality (not too strong nor too weak): $T_x = 1$, $T_y = 0.1$, and $T_v = 1.0$. We discuss more about the empirical aspect of the PFCC's parameters in Section 5.4.2. For HFCR, IPFC, and Fuzzy CoDoK, we applied the following parameter settings, which were found using the procedure we explained in Section 3.6.1 of Chapter 3: $T_u = T_v = 0.001$ for HFCR, $T_u = T_v = 5$ for

IPFC, and $T_u = 0.25$, $T_v = 2$ for Fuzzy CoDoK. We set $\epsilon = 10^{-5}$ and $\tau_{max} = 200$ for this experiment.

For ease of evaluation, we divide the D_{o2} 's matrix into four row regions namely: $srow_1, \dots, srow_4$ and four column regions namely: $scol_1, \dots, scol_4$ as in Figure 5.4. The row and column membership distributions should be uniform respectively on each of these row and column regions. For this reason, in Tables 5.8 and 5.9, rather than presenting every individual membership, we show the averages of the u_{ci} memberships of all the rows in each row regions, and the averages of the v_{cj} memberships of all the columns in each column regions.

Table 5.8: The Object Memberships of D_{o2}

Object	PFCC's x_{ci}			HFCR's u_{ci}			IPFC's u_{ci}			Fuzzy CoDoK's u_{ci}		
	$\sum_{c=1}^C x_{ci}$			$\sum_{c=1}^C u_{ci}$			$\sum_{c=1}^C u_{ci}$			$\sum_{c=1}^C u_{ci}$		
	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3
$srow_1$	0.46	0.23	0.69	0.96	0.04	1	0.99	0.01	1	0.95	0.05	1
$srow_2$	0.52	0.52	1.04	0.5	0.5	1	0.5	0.5	1	0.5	0.5	1
$srow_3$	0.23	0.46	0.69	0.06	0.96	1	0.01	0.99	1	0.95	0.05	1
$srow_4$	0.1	0.1	0.2	0.5	0.5	1	0.5	0.5	1	0.5	0.5	1

Table 5.9: The Feature Memberships of D_{o2}

Feature	PFCC's v_{cj}			PFCC's y_{cj}			HFCR's v_{cj}			IPFC's v_{cj}			Fuzzy CoDoK's v_{cj}		
	$\sum_{c=1}^C v_{cj}$			$\sum_{c=1}^C y_{cj}$			$\sum_{c=1}^C v_{cj}$			$\sum_{c=1}^C v_{cj}$			$\sum_{c=1}^C v_{cj}$		
	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3
$scol_1$	0.83	0.17	1	0.0014	0.0012	0.0026	0.9	0.1	1	0.9	0.1	1	0.002	0	0.002
$scol_2$	0.5	0.5	1	0.0016	0.0016	0.0032	0.5	0.5	1	0.5	0.5	1	0.002	0.002	0.004
$scol_3$	0.17	0.83	1	0.0012	0.0014	0.0026	0.1	0.9	1	0.1	0.9	1	0	0.002	0.002
$scol_4$	0.5	0.5	1	0.0007	0.0007	0.0014	0.5	0.5	1	0.5	0.5	1	0	0	0

From Table 5.8, we can see that all four algorithms agree that the objects in $srow_1$ should exclusively belong to C_1 . This can be seen from the large x_{C_1i} values and low x_{C_2i} assigned by PFCC, and the large u_{C_1i} values and low u_{C_2i} assigned by HFCR, IPFC, and Fuzzy CoDoK for objects in $srow_1$. The same thing can be said for objects in $srow_3$ and co-cluster C_2 . The difference between PFCC and the three fuzzy co-clustering algorithms lies in how PFCC treats the objects in the overlapping region in $srow_2$ and the object outliers in $srow_4$. HFCR, IPFC, and Fuzzy CoDoK, due to the constraint imposed on their object partitioning memberships, assign the same values to all the object memberships in these two row regions, i.e. $u_{C_1i} = u_{C_2i} = \frac{1}{C} = 0.5$. This is to indicate that these objects have neutral influence to the two co-clusters. Intuitively however, outliers like the objects in $srow_4$ should have lower membership values to the two co-clusters. Such a representation can be captured by the object possibilistic membership x_{ci} of PFCC. It can be seen from Table 5.8 that the value of $\sum_{c=1}^C x_{ci}$ in PFCC is relatively smaller for objects in $srow_4$, as compared to the other objects in the matrix, including those in $srow_2$. By minimizing the impact of these outliers, PFCC should give more robust performance than HFCR, IPFC, and Fuzzy CoDoK when applied on datasets containing many object outliers.

Let us now consider the feature memberships in Table 5.9. We can observe that the two types of feature membership in PFCC are exactly the combination of the feature memberships of HFCR/IPFC and Fuzzy CoDoK. By relying on its feature partitioning membership v_{cj} alone, PFCC cannot distinguish between features in the overlapping region in $scol_2$ and feature outliers in $scol_4$. This is evident from the exactly same values of v_{cj} for the features in these two column regions. The feature ranking membership y_{cj} however, enables the algorithm to detect the outliers. We can see that we have the value of $\sum_{c=1}^C y_{cj}$ for the features in $scol_4$ is significantly smaller than the other features in the matrix. Assigning low membership values to feature outliers minimizes

their influence in the overall co-clustering process minimum. This in turn can ensure a better performance on datasets containing many feature outliers.

To conclude, with its object possibilistic membership x_{ci} and feature ranking membership y_{cj} , PFCC should allow better robustness to outliers in a high-dimensional context, as compared to HFCR, IPFC, and Fuzzy CoDoK. Again, based on the results we get, we could say that PFCC manages to achieve 100% accuracy in robustness to outliers because it can detect all the outliers present, both from the object and the feature perspective. On the other hand, HFCR, IPFC, and Fuzzy CoDoK cannot detect any existence of the outliers in the dataset.

5.4.2 Document Clustering

In this section, we apply PFCC on the same large benchmark document datasets as in Section 3.6.2 of Chapter 3. These datasets are detailed in Table 3.5 of Chapter 3. For comparisons, experimental results of DFPC, HFCR, and Fuzzy CoDoK [75] are also shown. First, we discuss some details on the implementations. These are then followed by the results and discussions.

Implementation Details

In this experiment, except for the setting of the parameters T_x , T_v , and T_y in PFCC, we applied exactly the same implementation procedures and parameter settings as in the document clustering experiment of DFPC found in Section 4.4.2 of Chapter 4.

To decide the optimum values of T_x , T_v , and T_y in PFCC, we adopted a similar empirical procedure as in DFPC (Section 4.4.2). We run a number of simulations with different parameter settings and selecting the one that results in the optimum performance. The following is the guideline we adopt to get the optimum parameters.

First we set the value of all three parameters to be high. As discussed in Section 5.2.1, with this initial setting no co-cluster is formed because the value of x_{ci} is too small and the values of v_{cj} and y_{cj} are too fuzzy. We then gradually reduce the value of T_v to increase the influence of the fuzzy partitioning membership v_{cj} . After the value of T_v falls below a certain threshold (which generally depends on the nature of a dataset), v_{cj} becomes too crisp, resulting in a significant drop in performance. At this point, we stop reducing T_v and select its value that gives the best performance so far. With T_v fixed at its optimum value, we then decide the optimum values of T_x , which influences the value of the possibilistic membership x_{ci} , and T_y , which influences the value of the fuzzy ranking membership y_{cj} . Again we gradually reduce the values of the two parameters. After a certain point, the values of these two parameters become too low, and cause a significant drop in the performance, due to the tendency to generate coincident clusters. At this point, we stop reducing T_x and T_y , and select the values that give the best performances. The optimum parameters for PFCC are listed in Table 5.10.

Table 5.10: Document Clustering Parameters Setting

Dataset	PFCC		
	T_x	T_y	T_v
<i>Classic3</i>	0.001	0.001	0.001
<i>Binary</i>	0.001	0.001	0.001
<i>Multi5</i>	0.001	0.002	0.01
<i>Reuters3</i>	0.001	0.001	0.01
<i>SS</i>	0.001	0.001	0.01
<i>SM</i>	0.001	0.001	0.01
<i>ABJ</i>	0.001	0.0015	0.01
<i>CJB</i>	0.001	0.001	0.1
<i>CAB</i>	0.001	0.002	0.1
<i>CB</i>	0.001	0.001	0.01
<i>WPD6s</i>	0.001	0.002	0.01

Similar to the case of DFPC, we observe that there are certain ranges of T_x , T_v , and

T_y in which PFCC can give optimum performances. These ranges generally correspond to the conditions where we have moderate levels of fuzziness from v_{cj} and y_{cj} (i.e. not too crisp nor too fuzzy), and moderate level of typicality from x_{ci} (i.e. not too strong nor too weak). Within the boundaries of these ranges, PFCC's performances are not affected very much by the T_x , T_v , and T_y parameter settings. In the future, we may consider exploring some techniques that can enable PFCC to decide its own optimum parameter setting.

To evaluate the document clustering performance of PFCC, each document was assigned to the cluster with the largest x_{ci} value. Based on this assignment, the precision, recall, and purity measures were calculated [29, 89]. The formulas for these three measures can also be found in Section 3.6.1 of Chapter 3.

Results and Discussions

Table 5.11 show the performances of PFCC, HFCC, Fuzzy CoDoK and DFPC. The standard deviations are shown in brackets. The corresponding bar chart comparing the performances of PFCC, HFCC, and Fuzzy CoDoK is shown in Figure 5.5. In this chart, the vertical axis gives the 10-test run average of *precision + recall + purity*. From the figure, we can observe some important points. First, except on the relatively simple *Classic3* where all algorithms can perform well, PFCC consistently outperforms Fuzzy CoDoK on all the datasets. This suggests an improvement from the existing FCCM model, which Fuzzy CoDoK represents. Second, between PFCC and HFCC, generally the former gives either better or comparable performances. On *SS* however, we can see that PFCC's accuracy falls slightly below that of HFCC. Several things may contribute to this. We note that PFCC relies on feature ranking membership instead of feature possibilistic membership to achieve robustness to feature outliers. In Section 3.5 of Chapter 3, we discuss that a ranking membership has some representation-

related limitations that may affect the corresponding algorithm's performance. Another possible reason is that the incremental update of x_{ci} in PFCC can possibly cause some deviations that may not occur in the more conventional batch membership update. Future studies are needed to explore these issues further.

Table 5.11: Document Clustering Performance

Dataset	PFCC			HFCC			Fuzzy CoDoK			DFPC		
	Prec.	Rec.	Pur.	Prec.	Rec.	Pur.	Prec.	Rec.	Pur.	Prec.	Rec.	Pur.
<i>Classic3</i>	98.57 (0)	98.45 (0)	98.51 (0)	98.6 (0.1)	98.5 (0)	98.6 (0)	98.6 (0)	98.33 (0)	98.46 (0)	98.73 (0)	98.59 (0)	98.69 (0)
<i>Binary</i>	92.43 (3.9)	92.4 (3.9)	92.4 (3.9)	88.73 (15.4)	88.72 (15.4)	88.72 (15.4)	86 (14.2)	85.96 (15)	85.96 (15)	92.35 (11.5)	91.96 (11.5)	91.96 (11.6)
<i>Multi5</i>	90.6 (1.3)	91.2 (1.8)	91.2 (1.8)	90.5 (6.2)	90 (6.6)	90.04 (6.6)	88.95 (11.3)	88.4 (11.1)	90.69 (11.1)	90.97 (6.9)	90.87 (6.6)	91.76 (7.8)
<i>Reuters3</i>	86.45 (1.4)	86.53 (1.8)	86.41 (1.5)	80.76 (4.3)	80.76 (4.7)	80.88 (4)	82.32 (10.2)	81.46 (11.1)	82.55 (9.9)	84.52 (2.6)	84.29 (2.5)	83.73 (4)
<i>SS</i>	88.03 (0)	88 (0.1)	88 (0.1)	88.94 (0)	88.9 (0.1)	88.9 (0.3)	82.01 (8.8)	81.44 (8.7)	81.44 (8.7)	89.12 (0.1)	89.05 (0)	89.05 (0)
<i>SM</i>	85.85 (0.8)	84.64 (0.7)	84.64 (0.7)	85.49 (3.4)	85.09 (3.3)	85.09 (3.3)	75.99 (5.2)	74.34 (3.7)	74.34 (3.7)	82.76 (2.2)	82.45 (2.2)	82.45 (2.2)
<i>ABJ</i>	85.46 (0)	84.73 (0)	84.73 (0)	85.18 (0)	84.8 (0)	84.8 (0.4)	83.61 (1.6)	78.26 (1.7)	78.26 (1.7)	86.83 (0.3)	86.54 (1.1)	86.54 (0.4)
<i>CJB</i>	88.64 (0.2)	89.9 (0.3)	89.5 (0.2)	88.36 (6.4)	88.91 (6.3)	89 (6.3)	89.37 (1.9)	87.5 (1)	88.82 (1.3)	90.45 (1.1)	91.73 (1)	91.36 (1)
<i>CAB</i>	84.42 (0)	83.03 (0)	86.18 (0)	83.2 (0.4)	83.4 (0.3)	85.7 (0.3)	79.95 (2.1)	78.83 (1.2)	80.35 (3)	84.42 (0.4)	84.47 (0.6)	86.6 (0.5)
<i>CB</i>	69.1 (2.7)	69 (2.7)	69 (2.7)	67.2 (2.8)	66.1 (2.7)	66.1 (2.7)	67.76 (5.7)	67.72 (5.6)	67.72 (5.6)	68.32 (0.9)	68 (0.8)	68 (0.9)
<i>WPD6s</i>	81.22 (3.9)	76.85 (5.4)	76.85 (5.4)	78.69 (4.0)	75.75 (3.9)	75.75 (3.9)	63.42 (5.5)	65.4 (4.6)	65.4 (4.6)	80.99 (2.9)	73.27 (2.8)	73.27 (2.8)

Figure 5.5: Performance Comparison of PFCC, HFCR, and Fuzzy CoDoK

Figure 5.6 shows the bar chart comparing the performances of PFCC and DFPC. From the figure, we can see that despite its relatively simpler formulation, PFCC is able to outperform DFPC on some datasets like *Reuters3*, *WPD6s*, *SM*, and *CB*. The main advantage of PFCC over DFPC however, comes from its faster real-time computation as discussed in Section 5.3. We measured the average real computational times for one iteration in PFCC and DFPC. Table 5.12 shows the results for all the datasets. Evidently from the table, PFCC is more efficient than DFPC. However, despite all these positive points about PFCC, the algorithm should only be effective in a specific domain where Assumption 1 holds. For any other domain of application, DFPC is a better option for fuzzy-possibilistic co-clustering.

We now demonstrate the improved interpretability brought about by the possibilistic document membership x_{ci} of PFCC. Following the illustration in Section 4.4.2 of Chapter 4, let us consider a few documents that score the lowest $\sum_{c=1}^C x_{ci}$ in *Classic3*

Figure 5.6: Performance Comparison of PFCC and DFPC

Table 5.12: Comparison of Real-time Performance per Iteration

Dataset	PFCC (ms)	DFPC (ms)
<i>Classic3</i>	176	321
<i>Binary</i>	42	82
<i>Multi5</i>	51	104
<i>Reuters3</i>	78	148
<i>SS</i>	350	647
<i>SM</i>	297	541
<i>ABJ</i>	1031	1866
<i>CJB</i>	666	1209
<i>CAB</i>	706	1239
<i>CB</i>	277	482
<i>WPD6s</i>	370	613

dataset. Figure 5.7 shows the full contents of the bottom four documents $doc_1 - doc_4$.

Figure 5.7: Bottom 4 *Classic3* Documents with the lowest PFCC's $\sum_{c=1}^C x_{ci}$ values

Compared to the more common lengthier documents in *Classic3*, e.g. doc_5 in Figure 5.8, documents $doc_1 - doc_4$ are merely short snippets, and they arguably exhibit more outlier-like quality than doc_5 . With possibilistic memberships, the distinction between these two groups of documents with different characteristics is made clear from the value of $\sum_{c=1}^C x_{ci}$, as shown in Table 5.13 (i.e. documents like doc_5 generally have large $\sum_{c=1}^C x_{ci}$ values).

Contrast the PFCC possibilistic memberships in Table 5.13 with the corresponding HFCR fuzzy partitioning memberships shown in Table 5.14. Quite obviously, the partitioning memberships are not as descriptive as the possibilistic counterparts.

Table 5.15 shows the top words with the highest membership values in the five clusters (or co-clusters) of *Multi5* generated by PFCC, HFCR, and Fuzzy CoDoK.

Table 5.13: PFCC Possibilistic Memberships of *Classic3*'s doc_1 - doc_5

Document	C_1	C_2	C_3	$\sum_{c=1}^C x_{ci}$
doc_1	0.19	0.02	0.23	0.44
doc_2	0.174	0.288	0.03	0.492
doc_3	0.39	0.06	0.05	0.5
doc_4	0.04	0.44	0.04	0.52
doc_5	1.69	0.23	0.26	2.18

doc5:**Cost Analysis and Simulation Procedures for the Evaluation of Large Information Systems**

A computer program has been written and used which simulates the several-year operation of an information system and computes estimates of the expected operating costs as well as the amount of equipment and personnel required during that time period. The program has been used for the analysis of several large systems and has proven itself to be a useful research tool for the study of systems with so many components and interrelated operations that an equivalent manual analysis would be extremely cumbersome and time consuming, and perhaps even impractical. This paper describes this program and shows, as an example, some of the results of a simulation of two of several suggested designs for a specific information system.

Figure 5.8: Common *Classic3* DocumentsTable 5.14: HFCR Partitioning Memberships of *Classic3*'s doc_1 - doc_5

Document	C_1	C_2	C_3	$\sum_{c=1}^C u_{ci}$
doc_1	0.98	0	0.02	1
doc_2	0	1	0	1
doc_3	1	0	0	1
doc_4	0	1	0	1
doc_5	1	0	0	1

Table 5.15: *Multi5*'s Top Words

Cluster	PFCC ($v_{cj}, \frac{y_{cj}}{10^{-2}}$)	HFCR (v_{cj})	Fuzzy CoDoK ($\frac{v_{cj}}{10^{-2}}$)
C_1 (space)	space (1.0, 5.9) nasa (0.999, 0.79) pat (0.998, 0.52) comet (0.998, 0.41) orbit (0.986, 0.31) wings (0.975, 0.21) fred (0.97, 0.17) flight (0.955, 0.16)	space (0.999) nasa (0.998) pat (0.997) comet (0.996) orbit (0.99) fred (0.982) station (0.977) nuclear (0.956)	space (2.48) nasa (1.37) pat (1.28) comet (1.17) orbit (1.09) fred (0.95) station (0.94) nuclear (0.79)
C_2 (computer graphics)	graphics (0.999, 1.09) polygon (0.989, 0.27) algorithm (0.98, 0.23) code (0.959, 0.17) vga (0.954, 0.13) pov (0.939, 0.12) drivers (0.938, 0.12) software (0.27, 0.1)	graphics (1.0) polygon (0.994) algorithm (0.991) code (0.978) file (0.962) pov (0.961) vga (0.955) image (0.954)	graphics (1.99) polygon (1.21) algorithm (1.2) software (1.16) file (1.05) computer (1.0) code (1.0) pov (0.93)
C_3 (motorcycles)	bike (0.996, 0.85) jody (0.982, 0.24) ride (0.975, 0.22) riding (0.972, 0.2) dod (0.966, 0.22) drink (0.953, 0.15) levine (0.95, 0.14) drinking (0.857, 0.09)	bike (1.0) dod (0.994) riding (0.991) ride (0.989) jody (0.986) levine (0.97) drink (0.957) max (0.949)	bike (2.09) dod (1.42) riding (1.11) ride (1.04) jody (1.04) levine (0.84) bmw (0.8) max (0.76)
C_4 (baseball)	year (0.998, 1.61) players (0.998, 0.62) baseball (0.997, 0.47) season (0.992, 0.33) hitter (0.988, 0.28) bob (0.988, 0.25) win (0.987, 0.23) games (0.984, 0.21)	players (0.997) baseball (0.994) year (0.992) season (0.981) win (0.981) hitter (0.981) game (0.966) games (0.964)	year (1.34) players (1.31) baseball (1.12) game (1.06) season (0.94) win (0.91) hitter (0.9) bob (0.85)
C_5 (middle east)	armenian (1.0, 9.23) israeli (1.0, 3.75) turkish (1.0, 2.56) armenians (1.0, 1.98) israel (1.0, 1.71) people (0.999, 1.59) armenia (0.999, 0.72) killed (0.998, 0.47)	armenian (1.0) israeli (1.0) turkish (1.0) israel (1.0) armenians (0.999) armenia (0.998) arabs (0.996) people (0.994)	israeli (1.89) armenian (1.56) israel (1.54) turkish (1.45) armenians (1.18) people (1.11) armenia (1.1) arabs (1.01)

Since PFCC has two types of word membership, i.e. v_{cj} and y_{cj} , the values of $v_{cj} + y_{cj}$ were used to decide the top words. It can be observed from the table that these words can generally describe the contexts of their corresponding document clusters, e.g. from words such as “graphics” and “algorithm”, one can deduce that computer graphics is the main context of document cluster C_2 . Similar to DFPC, PFCC provides two types of meaningful information, i.e. how the word is partitioned (v_{cj}) and how the word is ranked in the clusters (y_{cj}). In this sense, we can say that PFCC is more informative than fuzzy co-clustering algorithms like HFCR or Fuzzy CoDoK, which produce only one type of word membership.

Between PFCC and DFPC, although they both generate two types of word memberships, PFCC computes word ranking membership rather than word possibilistic membership as in the case of DFPC. In Section 4.2 of Chapter 4, we have discussed how a possibilistic membership can be a more natural representation of word distribution than a ranking membership. This difference does not seem to affect the two algorithms’ selections of top words, if we consider the results in Table 5.15 here and Table 4.12 in Section 4.4.2 of Chapter 4. However, our study in Chapter 6 on the utilization of DFPC and PFCC’s word memberships to generate document clusters summaries seems to support our hypothesis that DFPC, due to its possibilistic word membership, can represent word membership better than PFCC.

Lastly, we discuss the validity of Assumption 1 in all our experiments, including the one discussed in Section 5.4.1. Assumption 1 states that at any point in the optimization of PFCC, we have $x_{cb} \ll \sum_{i=1}^N x_{ci}$. During the experiments, we calculated the ratio of $\frac{x_{cb}}{\sum_{i=1}^N x_{ci}}$ for every update, i.e. for every cluster c , object/document i , and iteration τ . Then we recorded the largest of such ratios (i.e. the worst case with regard to Assumption 1) out of all the updates that took place in every dataset. Mathematically this largest ratio is denoted by $\max \left\{ \frac{x_{cb}(\tau)}{\sum_{i=1}^N x_{ci}(\tau)} \mid \forall c, b, \tau \right\}$. The following Table 5.16 shows

the findings:

Table 5.16: Validity of Assumption 1

Dataset	No. of Docs (N)	$\max \left\{ \frac{x_{cb}(\tau)}{\sum_{i=1}^N x_{ci}(\tau)} \mid \forall c, b, \tau \right\}$
<i>2-D dataset</i>	602	0.0043
<i>D_{o2}</i>	600	0.0048
<i>Classic3</i>	3891	0.0011
<i>Binary</i>	500	0.0066
<i>Multi5</i>	500	0.013
<i>Reuters3</i>	1076	0.0033
<i>SS</i>	2000	0.0019
<i>SM</i>	2000	0.0017
<i>ABJ</i>	3000	0.0013
<i>CJB</i>	2250	0.002
<i>CAB</i>	2250	0.002
<i>CB</i>	2000	0.002
<i>WPD6s</i>	600	0.01

The figures in the last column of the above table reveal that the algorithm maintains $x_{cb} \ll \sum_{i=1}^N x_{ci}$ throughout its optimization process in all the datasets. This indicates that Assumption 1, to a large degree, remains valid in all our experiments on large datasets. The larger the sizes of the clusters in the datasets (e.g. the *Classic3* dataset), the more valid Assumption 1 is (i.e. the smaller the $\frac{x_{cb}}{\sum_{i=1}^N x_{ci}}$ ratios). The largest $\max \frac{x_{cb}}{\sum_{i=1}^N x_{ci}}$ value we got is from the *Multi5* dataset, which has clusters containing 100 documents each (i.e. the smallest cluster of all the datasets in Table 3.5). Even in this case, the value of x_{ci} is still less than 1.5 % of the value of $\sum_{i=1}^N x_{ci}$. Nevertheless, we should be cautious when dealing with smaller size clusters as the accuracy of PFCC can be compromised in this case.

5.5 Conclusions

We have presented PFCC, a new and simple fuzzy-possibilistic co-clustering model for large and high-dimensional data analysis. Just like DFPC, which is discussed in Chapter 4, PFCC offers important benefits such as robustness to outliers and rich representations of co-clusters due to the incorporated fuzzy and possibilistic memberships. While the new algorithm's applicability is limited to discover large size clusters, it offers a more efficient runtime than DFPC, due to its simple formulation that requires fewer computations. Through experiments, including on eleven large benchmark document datasets, we have demonstrated the effectiveness of PFCC as compared to its related fuzzy co-clustering counterparts. The experiments also show how PFCC, within its domain of application, can perform comparably with DFPC. In Chapter 6, we explore further the interpretability aspect of PFCC, by generating document clusters summaries based on the algorithm's word clusters.

There are some potential research directions. Most of them are similar with the future directions of DFPC: development of techniques to automatically decide the optimum parameters, formulation of a more flexible PFCC's objective function that allows many-to-many or correspondence between object and feature clusters (i.e. the number of object clusters can be different from the number of feature clusters); and incorporation of a dynamic weighting mechanism to adjust the importance of feature partitioning and feature ranking memberships at any time during the co-clustering process. Lastly, we note that PFCC trades off simplicity and efficiency for a limited domain of applications. In the future, we should aim for a simple and efficient fuzzy-possibilistic co-clustering that can be applied for any general application.

Chapter 6

Summarization of Document Clusters

An Application of Fuzzy-possibilistic Co-clustering

6.1 Overview

In this chapter, we discuss one application of fuzzy-possibilistic co-clustering for multi-document summarization (MDS) purpose. An MDS engine is used to automatically generate a brief and informative summary of a document collection. Here we consider the task of summarizing document clusters, as one possible form of document collections. By attaching a description summary on every generated document cluster, we can make the outcome of a clustering algorithm more interpretable and interesting. The objective of this chapter is therefore to demonstrate the potential of generating descriptive document clusters summaries based on our fuzzy-possibilistic co-clustering algorithms.

In the framework of co-clustering, the concept of automatic generation of docu-

ment clusters summaries comes naturally due to the process of word clustering that takes place simultaneously and interdependently with the process of document clustering. Because of this, we expect every resulting word cluster to indicate the content of its corresponding document cluster (i.e. most co-clustering establish such a correspondance) [29]. Furthermore, by utilizing soft word memberships in fuzzy-possibilistic co-clustering, we provide the algorithm a mean to represent the content more naturally. We argue later in Section 6.3.1 that there are some useful informations about document clusters that are exclusively contained in the associated soft word memberships.

This motivates us to enhance an existing MDS engine and come up with a new *word membership-based* MDS engine, in which, the word memberships of our fuzzy-possibilistic co-clustering algorithms are used as the main inputs. For this purpose, we choose to use MEAD [106], which is a well-known summarizer that generates summaries by extracting salient sentences from a document collection. Through simulations on a benchmark dataset, we show that the word membership-based MEAD can outperform the original MEAD for the purpose of document clusters summarization.

We organize this chapter as follows. In Section 6.2, we review some existing works in the MDS field. In Section 6.3, we discuss the new word membership-based summarizer for summarization of document clusters, including the motivation behind and some simulations on a benchmark document dataset. This chapter is concluded in Section 6.4.

6.2 Literature Review

We first review some existing major approaches to MDS. This is followed by a more detailed review of a particular MDS technique called MEAD [106], which is used as the base for our word membership-based summarizer. We then discuss ROUGE [83], which

is one prominent evaluation measure for MDS.

6.2.1 Multi-document Summarization Review

There are two main techniques of summarization: extraction and abstraction. In the former, the generated summaries are in the form of salient information, e.g. sentences, extracted from documents; while in the latter, the generated summaries are synthesized based on the contextual and linguistic understandings of the summarizers. Majority of multi-document summarizers available in the literature are extraction-based summarizers [135], and therefore their goal is to extract the most salient information from a document collection. There are many different approaches to decide the saliency of information. In the following, we highlight some of the major approaches, along with some examples of algorithm:

- *Statistical Approach:* Ideally, an automatic summarizer, regardless of it is being an extraction or abstraction-based, should have some language understanding capabilities. At the current state however, the natural language processing technology is not yet practical enough for general real-world applications [10]. Hence, many summarizers still rely on the statistical approach, in which word or phrase distributions are keys to decide saliency of information. Examples of these approaches include: MEAD [106], MMR [15], SumBasic [98] and its variants in [99, 130], and NMF-based summarizer [104].
- *Structural Approach:* In this approach, saliency of information is decided through a semantic analysis of the structural representation of documents. Two examples of algorithm that adopts this approach can be found in [145], which uses the lexical chain structure; and in [139], which uses the document concept lattice structure.

- *Linguistic Approach:* Some summarizers have incorporated linguistic analysis to a certain degree to capture salient information. Some examples include: LAKE [25], Summarization based on topic and theme representations [55], and a summarizer using language modeling [62].
- *Other Approach:* There are many other approaches to extraction-based MDS. Examples are: Genetic algorithm-based MDS called MSBGA [60], MDS using PageRank [12], and MDS with supervised perceptron-based ranker [40].

Our word membership-based summarizer is essentially an enhanced version of the MEAD summarizer. Therefore, it can be categorized under the statistical approach to extraction-based MDS. Previous works have never considered using fuzzy or possibilistic word memberships to decide the saliency of information. As discussed in Chapters 4 and 5, the way we formulated the soft word memberships in our fuzzy-possibilistic co-clustering algorithms can offer realistic and robust representations of words importance in document clusters. Hence they should provide useful inputs for an MDS engine.

6.2.2 The MEAD Summarizer

We now discuss the MEAD summarizer in more details. As indicated earlier, MEAD provides the foundation of our word membership-based approach to MDS. There are two main modules in MEAD: document categorizer and document summarizer. MEAD first applies document clustering, followed by a summarization of the resulting document clusters. MEAD applies a centroid-based technique to cluster documents. In the original version of MEAD, a centroid is a vector in a space made up by important words in a cluster. Each element of a centroid vector is represented by the *tf-idf* [111] value of every word in a cluster. The clustering algorithm starts with a centroid computed from a single document. It then proceeds iteratively to evaluate every other document in

the corpus. In each iteration, the cosine similarity between a cluster centroid and the document being evaluated is calculated. If the similarity is above a certain threshold, the document is added into the cluster and the centroid is updated.

To generate a summary of a document cluster, MEAD employs a sentence ranking technique. The top few sentences in a document cluster constitute the summary of the cluster. There are three main factors to consider in deciding the rank of a sentence: centroid value, positional value, and first-sentence-overlap value [106]. They are respectively defined in Eqs. (6.1)-(6.3), with NS denotes the number of sentences in a document and $sent_1$ denotes the first sentence in a document. A centroid value $cent_{c,sent}$ captures the similarity between a sentence $sent$ and a cluster centroid c . A positional value $pos_{c,sent}$ assumes that the earlier a sentence appears in a document, the more important it is. A first-sentence-overlap value $fso_{c,sent}$ measures the similarity between a sentence and the first sentence of the same document. The sentence score $score(sent)$ is then calculated using Eq. (6.4), where w denotes the weights to be defined manually. In addition to computing this score, MEAD employs an additional iterative procedure to re-rank sentences based on their information redundancy (i.e. CSIS) measures. This last step is important to reduce the amount of repetition found in the resulting summaries. Some other factors such as sentence length can also be taken into account in determining the sentence score [105].

$$cent_{c,sent} = \sum_{word \in sent} (tf-idf)_{c,word} \quad (6.1)$$

$$pos_{c,sent} = \frac{NS - sent + 1}{NS} \max_{sent \in c} cent_{c,sent} \quad (6.2)$$

$$fso_{c,sent} = \sum_{word \in c} (tf-idf)_{sent_1, word} (tf-idf)_{sent, word} \quad (6.3)$$

$$score(sent) = w_c cent_{c,sent} + w_p pos_{c,sent} + w_f fso_{c,sent} \quad (6.4)$$

6.2.3 ROUGE Evaluation Measure

One popular evaluation measure for summarization is Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [83]. The ROUGE score of an automatically generated summary is measured based on the consistency between its content and the content of a reference summary (or summaries). The latter is normally produced by human experts. We employ three commonly used versions of ROUGE in our simulations: ROUGE-N, ROUGE-L, and ROUGE-W [102].

ROUGE-N is essentially an N-gram recall measure between the resulting summary and the corresponding reference summary. The logic is that the more N-grams in the resulting summary matched with the N-grams in the reference summary, the likelier the two summaries share similar contents. The mathematical definition of ROUGE-N is as follow:

$$ROUGE-N = \frac{\zeta_N}{\zeta_N + \xi_N}$$

where ζ_N denotes the number of N-grams occur in both the resulting and reference summaries, while ξ_N denotes the number of N-grams occur in the reference summary but not in the resulting summary. In our demonstration in Section 6.3.3, we use $N = 1$ and 2 (i.e. ROUGE-1 and ROUGE-2), which are the popular selections in the field [99, 60, 139, 55].

ROUGE-L captures the summary-level longest common subsequence (LCS) between the resulting and reference summaries. Unlike ROUGE-N, ROUGE-L takes into account the sequence of the words appearing in a sentence. ROUGE-L is mathematically defined as follows:

$$ROUGE-L = \frac{(1 + \beta^2) R_{rouge} P_{rouge}}{R_{rouge} + \beta^2 P_{rouge}}, \text{ where:}$$

$$R_{rouge} = \frac{\sum_i |\cup_j LCS(r_i, s_j)|}{m} \text{ and } P_{rouge} = \frac{\sum_i |\cup_j LCS(r_i, s_j)|}{n}$$

In the above, m and n denote the numbers of words in the reference and resulting summaries respectively; and $LCS(r_i, s_j)$ denotes the set of sentence-level LCS between the sentence r_i in the resulting summary and the sentence s_j in the reference summary. In our simulation in Section 6.3.3, we follow the specification in the ROUGE working note [84] and set $\beta = 0$.

ROUGE-W is the weighted version of ROUGE-L. One drawback of ROUGE-L is that it ignores any consecutive match in the two sentences being compared [83]. In computing the ROUGE-W score, this number of consecutive matches is taken into consideration. Computation of ROUGE-W requires a combination of dynamic programming and some additional calculations, as detailed in [83].

6.3 Word Membership-based MDS

In this section, we present one approach to generate descriptive document clusters summaries by integrating a fuzzy-possibilistic co-clustering algorithm with an existing MDS system. First we discuss the motivation behind such an approach. We then present a new word membership-based summarizer. Lastly, we demonstrate the new

MDS system by performing some experiments on a benchmark dataset.

6.3.1 The Motivation

The word memberships ($v_{cj} + y_{cj}$) generated by our fuzzy-possibilistic co-clustering algorithms DFPC and PFCC should serve as important inputs for document clusters summarization because they exclusively contain some useful information about the associated document clusters. Let us recall that a co-cluster in this context is an association between document cluster and word cluster. Both v_{cj} and y_{cj} of the two algorithms essentially measure the similarity between word j and co-cluster c , and therefore, the higher their values, the more should the word j capture the contents of the associated document cluster c . In addition, the possibilistic word membership y_{cj} in DFPC or the ranking membership y_{cj} in PFCC provides a measure of similarity between words and co-clusters that is robust to word outliers, that is, it can filter out any existing word outlier irrelevant to the document clusters' contents. This is further enhanced by the information provided by the fuzzy partitioning word membership v_{cj} in DFPC or PFCC, which indicates certain words that are uniquely-associated to a particular co-cluster, and consequently should serve as unique content representatives of the corresponding document cluster. We note that v_{ab} can have a high value iff word b has a high similarity to co-cluster a , but low similarities to all the other co-clusters (i.e. recall that the word partitioning membership v_{cj} is a measure of a word similarity to a particular cluster, relative to the word's similarity to all the other co-clusters). Finally, we note from Eqs. (4.4) and (4.6) in DFPC, or Eqs. (5.3) and (5.4) in PFCC, that v_{cj} and y_{cj} are both functions several variables, including the document memberships x_{ci} (and u_{ci} in the case of DFPC). We have argued in Chapters 4 and 5 that the document memberships of the two fuzzy-possibilistic co-clustering algorithms are highly realistic

and robust representations of document clusters. By taking them into consideration, the word memberships v_{cj} and y_{cj} can capture the states of document clusters to be summarized more closely.

All these important informations are exclusively and compactly contained in the soft word memberships of our fuzzy-possibilistic co-clustering algorithms. For this reason, they should serve as highly informative inputs for document clusters summarization.

6.3.2 The Word Membership-based MEAD

We modify the MEAD summarizer, first by replacing its clustering module with our fuzzy-possibilistic co-clustering algorithms (i.e. DFPC or PFCC), and second by incorporating the resulting word memberships as inputs to the system, substituting the original *tf-idf*-based input as explained in Section 6.2.2. We choose this substitution scheme, rather than incorporating the word memberships as new additional inputs to MEAD so that we can empirically show (in Section 6.3.3) that by using the word memberships alone as the main inputs, we can have improved summarization performances. We refer to the newly-modified summarizer simply as the *word membership-based MEAD*. We note that our selection of MEAD is primarily due to its easily modified formulation and its freely available and well-documented software package.

To incorporate the word memberships, we replace the original *tf-idf*-based centroid value in Eq. (6.1) with our *word membership-based* centroid value defined in Eq. (6.5) when computing the *score(sent)* in Eqs. (6.2) and (6.4) (we neglect the first-sentence-overlap factor in our implementation). In Eq. (6.5), $V_{cj} = v_{cj} + y_{cj}$ in the case of DFPC, and $V_{cj} = v_{cj} + w_{cj}$ in the case of PFCC. For DFPC, since the MEAD system recommends a positive sentence score, we standardized the y_{cj} membership by shifting its value to the positive quadrant for every c and normalized them against the maximum

membership in c .

$$cent_{c,sent} \equiv memb_cent_{c,sent} = \sum_{j \in sent} V_{cj} \quad (6.5)$$

Figure 6.1 shows an example of document clusters summaries generated by the word membership-based MEAD (in this case DFPC) on the *Classic3* dataset.

<p>Cluster 1 Summary:</p> <ol style="list-style-type: none"> 1. axisymmetric large deflections of circular plates subjected to thermal and mechanical load. 2. force-test investigation of the stability and control characteristics of a scale model of a tilt-wing vertical take-off and landing aircraft. 3. the transonic flow of a compressible fluid through an axially symmetrical nozzle. 4. an analysis of base pressure at supersonic velocities and comparison with experiment. 5. stability of compressible boundary layers induced by a moving wave. 6. the law of the wake in the turbulent boundary layer. 7. on flow of electrically conducting fluids over a flat plate in the presence of a transverse magnetic field. 8. a discussion of differences arising between experimental and analytical results, in particular those due to inconsistencies introduced in the presentation of data and the way the comparison is made. 9. flutter model testing at transonic speeds flutter research on reflection plane models of straight, swept, and delta wings in a 3 x 4 foot transonic test facility. 10. some current and proposed investigations into the flow for slender delta and other wings in unsteady motion. 	<p>Cluster 2 Summary:</p> <ol style="list-style-type: none"> 1. cortisone and atypical pulmonary /epithelial/ hyperplasia further studies including electron microscopy, tissue culture, animal transplantation and long term observations. 2. the behavior of lymphocytes in primary explants of human lung cancer in vitro. 3. radioisotope photoscanning as a diagnostic aid in cardiovascular disease a safe, simple method for diagnosing pericardial effusion and aneurysm . 4. studies on cardiac dimensions in intact, unanesthetized man. i. description of techniques and their validation. ii. effects of respiration. iii. effects of muscular exercise a method is described which permits measurement of relative changes in the external dimensions of individual cardiac chambers throughout the cardiac cycle in intact, unanesthetized man. 5. reactions of seleno and sulfoamino acids with hydroperoxides reactions of methionine, cystine and their selenium analogues with hydrogen peroxide and organic peroxides have been studied. 6. influence of hormones on tumor growth and plasma prolactin levels in rats bearing a pituitary "mammatropic" tumor the effects of estradiol, cortisol acetate, thyroxine, or thiouracil on tumor growth, organ weights, and plasma prolactin levels were determined in rats bearing a furth pituitary "mammatropic" tumor mtt.f4 . 7. dreaming sleep in autistic twins each of 5-year-old identical autistic twins was observed during one spontaneous all night sleep session during which their electroencephalogram, eye movement, heart rate, and chin muscle potential were recorded. these autistic twins showed a similar patterning and amount of dreaming sleep to each other and to that reported for normal children of the same age. 8. localization of phosphaturic effect of parathyroid hormone in nephron of the dog the renal mechanism and tubular site of the phosphaturic action of parathyroid hormone were investigated by clearance and stop-flow techniques in one intact and seven thyroparathyroidectomized dogs. 9. plasma growth hormone concentration in the rat determined by radioimmunoassay: influence of sex, pregnancy, lactation, anesthesia, hypophysectomy and extrasellar pituitary transplants. 10. kidney cell proliferation after unilateral nephrectomy as related to age.
<p>Cluster 3 Summary:</p> <ol style="list-style-type: none"> 1. Psychological Problems in Informatics and the Prospects of their Solution. 2. Out of the Dinosaur; the evaluation of the National Lending Library. 3. Evaluation of Methods of Formal Investigation of Texts in Dead Languages. 4. Cognitive, Technical and Social Factors in the Growth of Radio Astronomy We have tried in this paper to describe some of the main features of the emergence and growth of radio astronomy, with special reference to the crucial developments occurring in the UK. 5. A Bookshelf in Public Health, Medical Care, and Allied Fields. 6. Experimental Use of a Program for Computer-Aided Subject-Index Production. 7. Text Searching Retrieval of Answer-Sentences and Other Answer-Passages. 8. RELEVANCE: A Review of and a Framework for the Thinking on the Notion in Information Science Information science emerged as the third subject, along with logic and philosophy, to deal with relevance - an elusive, human notion. 9. A Statistical Interpretation of Term Specificity and Its Application in Retrieval The exhaustivity of document descriptions and the specificity of index terms are usually regarded as independent. 10. application of the emission spectrograph to the analytical needs of the industrial hygiene laboratory. 	

Figure 6.1: *Classic3* Document Clusters Summaries

6.3.3 Experiments

To demonstrate the potential of the word membership-based MEAD on the MDS task, we conducted some experiments on the Document Understanding Conference (DUC) 2004 [102] dataset. The availability of the ground-truth summarization of the DUC 2004

dataset makes it feasible to evaluate the performance of an MDS engine. We contrast the word membership-based and the original versions of MEAD to evaluate the effectiveness of the former. We tested several different versions of word membership-based MEAD; two of them correspond to the two fuzzy-possibilistic co-clustering algorithms we have, DFPC and PFCC. We refer to them as DFPC-MEAD and PFCC-MEAD respectively. To enrich the discussions, we also ran the third version of word membership-based MEAD based on Fuzzy CoDoK [75], i.e. FC-MEAD. Similar to the cases of DFPC-MEAD and PFCC-MEAD, in FC-MEAD, the clustering module of MEAD is replaced with Fuzzy CoDoK, and the resulting Fuzzy CoDoK’s fuzzy word memberships are used as inputs to the system as substitutes to the original *tf-idf*.

In the following, we provide the details of DUC 2004, as well as our implementations. This is then followed by the results and discussions.

Dataset and Implementation Details

The DUC 2004 dataset (or simply the DUC 2004) consists of 500 news articles, categorized into 50 groups of 10 articles. Although the dataset has already been organized manually into 50 different categories, the three versions of membership-based MEAD, i.e. DFPC-MEAD, PFCC-MEAD, and FC-MEAD, still require document clustering (or co-clustering) to be performed (respectively by applying DFPC, PFCC, and Fuzzy CoDoK), as these summarizers need the word memberships, which can only be made available by actually running the co-clustering processes. To simplify the document clustering process, considering some article groups contain very similar topics (e.g. both groups 7 and 9 discuss the trial of Anwar Ibrahim), we segregate the 500 articles into 10 subsets, each consists of 5 different article groups of total 50 articles. We arranged in such a way that each of the 10 subsets does not contain groups sharing very similar topics. Co-clustering was then performed on each subset to identify the 5 clusters it

contains. For the original MEAD, we directly used the manual categorization provided, i.e. its clustering module as explained in Section 6.2.2 was not employed at all.

Additionally, with only 10 documents in each cluster of the DUC 2004 dataset, the Assumption 1 in Section 5.2.2 of Chapter 5, which indicates that there should be a significantly large number of objects in every cluster for PFCC to run properly, does not hold. Hence, for our PFCC-MEAD demonstration on the DUC 2004 (only), we artificially increased the size of each document cluster in the dataset. This was done by generating 9 duplicates of each document in the DUC 2004, and then including them in the dataset. This increases the size of each cluster to 100 documents. It should not be too difficult to appreciate that the original DUC 2004 dataset and its expanded version (i.e. the one containing document duplicates) share the same inherent word clustering structure, or equivalently the same ideal word memberships. Hence, in the case of PFCC-MEAD, we should get the word memberships of the DUC 2004 dataset, which are to be used as inputs for summarization of the dataset, by performing co-clustering on the expanded DUC 2004 instead. This is because the Assumption 1 in the PFCC formulation is valid on the expanded version of the DUC 2004 dataset, in which we have a significantly large number of documents in every cluster.

The DUC 2004 dataset, and its expanded version in the case of PFCC-MEAD, were pre-processed using the MC toolkit following the same procedures discussed in the experimental results sections of Chapters 4 and 5. This time however, except for the stopwords, all the words were taken into account.

With all these preprocessing procedures, we found that all DFPC, PFCC, and Fuzzy CoDoK could achieve 100% accurate document clustering for all the corresponding datasets' subsets (i.e. as measured against the manually categorized groups). The followings are the parameter settings employed: $T_u = 0.5$, $T_v = 0.5$, $T_x = 5$, $T_y = 20$ for DFPC; $T_u = 0.01$, $T_y = 0.01$, $T_v = 0.05$ for PFCC; and $T_u = 0.0001$, $T_v = 1.5$ for Fuzzy

CoDoK. We followed the same procedures as discussed in Sections 3.6.1, 4.4.2, 5.4.2 to get these optimum parameter values.

We use four evaluation measures: ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-W (with $W = 1.2$, following the convention for the DUC 2004 dataset [102]). Each of the summarizers was set to generate for each document cluster, a description summary made of the top 20% sentences out of all the original documents' sentences in the cluster (i.e. excluding the sentences in the duplicate documents in the case of PFCC-MEAD). Two factors were considered in the summarization process: the centroid value (the membership-based in DFPC-MEAD, PFCC-MEAD, FC-MEAD; and the *tf-idf* in the original MEAD) and the sentence position. These two factors have been discussed in Section 6.2.2. We assigned the weights $w_c = 5$ and $w_p = 1$ for the centroid value and the sentence position respectively. By setting a higher weight for the centroid value, the word memberships become the main inputs to MEAD. This enables us to observe the impact of the word memberships on the resulting summaries. In our demonstrations, the original redundancy removal procedure in MEAD, as discussed in Section 6.2.2, was also performed.

Results and Discussions

Tables 6.1-6.4 show respectively the performances of MEAD, DFPC-MEAD, PFCC-MEAD, and FC-MEAD on all the 10 subsets of DUC 2004 (or its expanded version in the case of PFCC-MEAD). Figure 6.2 summarizes the averages the ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-W. We can see that DFPC-MEAD and PFCC-MEAD can outperform MEAD on all evaluation measures. Inline with our analysis in Section 6.3.1, this empirical finding suggests that the soft word memberships of DFPC or PFCC can serve as important inputs to an MDS engine. Our experiments specifically demonstrate that, in this case, the word memberships are the better inputs than the word

frequency-based *tf-idf* measures used in the original MEAD. This result can be important, especially in the context of document clusters summarization, as many existing summarization engines rely on word frequency measures to capture salient information [99, 45].

Another interesting observation is that, even though both DFPC-MEAD and PFCC-MEAD manage to generate better summaries than MEAD, between the former two, we can see that DFPC-MEAD performs better than PFCC-MEAD. This indicates that the kinds of word membership used to generate summaries matter. This is further supported by the results of FC-MEAD, which does not improve MEAD as much as DFPC-MEAD and PFCC-MEAD do on ROUGE-1, ROUGE-L, and ROUGE-W; and does not offer any improvement on ROUGE-2. We have argued in Chapters 4 and 5 that ideally fuzzy-possibilistic co-clustering algorithms like DFPC and PFCC should provide more informative outcomes (i.e. memberships) than fuzzy co-clustering algorithms like Fuzzy CoDoK. That FC-MEAD brings less improvement than the other two membership-based MEAD summarizers in this demonstration seems to agree with our proposition. One of the main differences between DFPC-MEAD and PFCC-MEAD lies in the fact that the former incorporates possibilistic word membership while the latter incorporates ranking word membership, in addition to their partitioning word memberships. We mention in Section 4.2 of Chapter 4 that a possibilistic membership can give a more natural representation of clusters than a ranking membership. This factor may contribute to DFPC-MEAD being able to generate better summaries than PFCC-MEAD.

While more extensive experiments may be needed to further support the findings we present in this section, our demonstration here can give initial insight on the potential of a word membership-based summarizer.

Figure 6.2: Summarization ROUGE Performance Comparison

Table 6.1: DPFC-MEAD Summarization Performance on DUC 2004

Subset	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W
1	0.779	0.266	0.741	0.312
2	0.762	0.253	0.743	0.319
3	0.748	0.207	0.721	0.299
4	0.769	0.278	0.732	0.307
5	0.773	0.284	0.735	0.311
6	0.788	0.29	0.762	0.328
7	0.771	0.273	0.75	0.325
8	0.781	0.234	0.757	0.334
9	0.78	0.247	0.741	0.313
10	0.761	0.292	0.735	0.317
Avg.	0.771	0.262	0.742	0.316

Table 6.2: PFCC-MEAD Summarization Performance on DUC 2004

Subset	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W
1	0.771	0.261	0.736	0.309
2	0.754	0.246	0.728	0.312
3	0.747	0.206	0.707	0.292
4	0.772	0.28	0.732	0.31
5	0.778	0.299	0.74	0.315
6	0.781	0.297	0.757	0.326
7	0.767	0.262	0.738	0.32
8	0.768	0.23	0.743	0.328
9	0.783	0.243	0.761	0.314
10	0.758	0.291	0.735	0.318
Avg.	0.768	0.261	0.738	0.314

6.4 Conclusions

We have presented one potential and practical application of fuzzy-possibilistic co-clustering for generating descriptive document clusters summaries. Such an application can make the outcomes of a document clustering algorithm more interpretable and interesting. We have discussed how the resulting word memberships of our fuzzy-possibilistic co-clustering algorithms can serve as important inputs to a multi-document summarization (MDS) system for document clusters summarization purpose. We have

Table 6.3: FC-MEAD Summarization Performance on DUC 2004

Subset	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W
1	0.765	0.259	0.73	0.306
2	0.767	0.266	0.737	0.318
3	0.721	0.213	0.697	0.287
4	0.762	0.269	0.735	0.308
5	0.765	0.281	0.722	0.304
6	0.772	0.285	0.745	0.32
7	0.763	0.266	0.742	0.32
8	0.787	0.241	0.75	0.329
9	0.779	0.242	0.741	0.312
10	0.73	0.264	0.703	0.302
Avg.	0.761	0.259	0.73	0.311

Table 6.4: MEAD Summarization Performance on DUC 2004

Subset	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W
1	0.758	0.261	0.725	0.303
2	0.763	0.27	0.733	0.314
3	0.744	0.21	0.713	0.295
4	0.755	0.271	0.718	0.303
5	0.754	0.282	0.708	0.299
6	0.768	0.283	0.748	0.323
7	0.774	0.275	0.751	0.324
8	0.774	0.246	0.739	0.335
9	0.775	0.241	0.735	0.31
10	0.706	0.253	0.685	0.294
Avg.	0.757	0.259	0.726	0.309

explained how we modify an existing MDS engine MEAD to come up with different versions of word membership-based summarizer, including those correspond to our two fuzzy-possibilistic co-clustering algorithms DFPC and PFCC. Our demonstration on a benchmark document dataset confirms the potentials of the fuzzy-possibilistic co-clustering algorithms for document clusters summarization task.

More extensive experiments studying the impact of using different types of word memberships on the summarization outcomes can be conducted in the future. Integrat-

ing the fuzzy-possibilistic co-clustering algorithms with other advanced summarization systems is also another interesting research direction.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

Data clustering is a crucial technique in the knowledge discovery process. Clustering can offer many potential benefits to wide-range of fields including information retrieval, bioinformatics, and business intelligence. In its current state, there are still many rooms to improve the quality of clustering. In this thesis, we have addressed several important aspects of clustering: high-dimensionality, quality of clusters representations, robustness to outliers, and interpretability of clusters. We have built a new clustering framework called fuzzy-possibilistic co-clustering to improve these four aspects simultaneously.

We have reviewed some major existing approaches to clustering. From the discussions, we have justified why we need a new fuzzy-possibilistic co-clustering framework for high-dimensional data analysis. We also have presented our detailed studies on some clustering models relevant to our work, i.e. Fuzzy C-means (FCM), Possibilistic C-means (PCM), Possibilistic-fuzzy C-means (PFCM), and Fuzzy Clustering for Categorical and Multivariate data (FCCM). We have discussed their advantages and disadvantages, and analyzed that, by combining the principles of these four models

together into a single formulation, we can form a new and better clustering framework.

With this goal in mind, we have conducted an in-depth study on one simple and yet efficient and practical fuzzy co-clustering model, the FCCM model. Our analysis, supported by an existing study, shows that, while generally more effective than its standard fuzzy clustering counterparts for high-dimensional data clustering, the FCCM model can still be improved in terms of robustness to object outliers and quality of clusters representations. Furthermore, the formulation of the existing model is so rigid, it only allows limited combinations of fuzzy memberships to be incorporated. This rigidity prevents us from expanding the FCCM model into fuzzy-possibilistic co-clustering, which theoretically can bring improvements on the aspects of robustness and representation capability.

To address the expandability issue in the FCCM model, we have proposed three new fuzzy co-clustering models, which can cater for a wider range of possible combinations of fuzzy memberships being incorporated. We have chosen to first develop the corresponding dual-partitioning approach to fuzzy co-clustering for each of these three models. The dual-partitioning approach to fuzzy co-clustering can be considered as the co-clustering version of Fuzzy C-means, which itself can be considered as the foundation of fuzzy clustering. For this reason, we treat our three dual-partitioning-based fuzzy co-clustering algorithms: FCR, HFCR, and IPFC; mainly as the foundation for fuzzy-possibilistic co-clustering. Additionally, we have analyzed the advantages and limitations of the dual-partitioning approach, as compared to the existing FCCM model. Consistent with this analysis, we have found from our experiments that while the dual-partitioning approach can generally perform well on high-dimensional datasets, there are some cases where it cannot perform as effectively than the FCCM model. This indicates the need for better extensions for the dual-partitioning approach. Among the three dual-partitioning models, we have shown both theoretically and empirically

that HFCR and IPFC are the better models than FCR. We have also explained that IPFC has a firmer theoretical basis than HFCR; but on the other hand, HFCR allows a simpler and more efficient fuzzy co-clustering than IPFC.

Experiments show some benefits of the dual-partitioning approach to fuzzy co-clustering. However, in our context, its most important benefit comes from the fact that the principles to formulate the approach can be extended to develop the more powerful fuzzy-possibilistic co-clustering framework. Hence, we have extended IPFC into DFPC, our first fuzzy-possibilistic co-clustering algorithm. With possibilistic memberships incorporated into its objective function, DFPC has better robustness to outliers than the dual-partitioning-based fuzzy co-clustering models, and also the existing FCCM model. Additionally, with its combined fuzzy and possibilistic memberships, DFPC allows more realistic and richer co-clusters representations than its fuzzy co-clustering counterparts. Experiments on document datasets confirm these. The experiments also show how DFPC consistently outperforms the existing FCCM model and the dual-partitioning approach when tested on several large document datasets. Furthermore, our experiments capture the effectiveness of DFPC for high-dimensional applications, as well as the algorithm's ability to automatically generate highly interpretable cluster descriptions.

Inspired by the simplicity of HFCR, which is one of our dual-partitioning-based fuzzy co-clustering algorithms, we have developed a new alternative HFCR-like fuzzy-possibilistic co-clustering algorithm PFCC, which has a simpler formulation than DFPC. Due to its simplicity, PFCC has a more efficient real-time computation than DFPC. This efficiency however, comes at the expense of the applicability of PFCC. The derivation of the PFCC's update rules is based on an assumption that datasets contain large size object clusters. We have argued that this is usually the case for large-scale datasets. Hence, PFCC can find its applications in the fields that involve categorizations of large-scale

datasets, e.g. categorization of articles in digital library, clustering of web documents for indexing, and classification of customer or product records in a data warehouse. We have shown through experiments on large document datasets that PFCC, within its scope of application, offers all the benefits of fuzzy-possibilistic co-clustering, including ability to perform well in high dimension, robustness to outliers, rich representations of clusters, and highly-interpretable clusters. Compared to DFPC, PFCC can achieve all these in faster real times, as demonstrated by our experiments. Performance comparisons with HFCR and Fuzzy CoDoK suggest that PFCC is generally more accurate than its fuzzy co-clustering counterparts.

We have also conducted some explorations on the ability of fuzzy-possibilistic co-clustering to generate highly interpretable clusters, in the context of automatic summarization of document clusters. We have discussed how existing multi-document summarizers can benefit from the resulting soft word memberships of our fuzzy-possibilistic co-clustering algorithms. Motivated by this, we have enhanced an existing multi-document summarizer, MEAD, by providing the soft word memberships of DFPC and PFCC as the summarizer's inputs. Demonstration on a benchmark dataset shows that the word membership-based MEAD is a better summarizer than its original counterpart. This finding suggests that fuzzy-possibilistic co-clustering can intrinsically produce highly interpretable descriptions of clusters.

Figure 7.1 summarizes the works that we have done in this project.

Figure 7.1: The Contributions of this Thesis

Finally, Table 7.1 shows some examples of practical implications and applications that may correspond to the unique features offered by the proposed fuzzy-possibilistic co-clustering. We note that it is possible to achieve these practical implications simultaneously.

Table 7.1: Practical Implications and Applications of Fuzzy-possibilistic Co-clustering

Features	Practical Implications and Applications
Dynamic dimensionality reduction	Effective categorization for analysis of high-dimensional real-world datasets. Examples: Clustering and analysis of web documents, gene expression data, business customers, and geographic datasets
Outlier detection ability	<ul style="list-style-type: none"> • Robust categorization of high-dimensional datasets in the presence of outliers. Example: Clustering of topics in an online discussion forum • Intrusion detection in a high-dimensional space. Example: Detecting a virus in a network security system • Novelty detection in a high-dimensional space. Example: Identifying new information in a patient's consultation notes
Modeling overlapping clusters	Realistic representation in the categorization of real-world datasets, which usually contain overlaps. Examples: Capturing overlaps in clusters of web search results, DNA microarrays, or potential markets
Generation of soft feature memberships	<ul style="list-style-type: none"> • Improved interpretability of clusters. Example: Automatic summarization of document clusters • Better insight to the clustering results by providing categorization of feature sets. Example: Generating a word thesaurus in the scope of a given document collection • Better insight to the clustering results by indicating which set of features are contributing to clustering of objects. Example: Identifying which conditions correspond to a cluster of genes

To conclude, the main contribution of this thesis lies in the development of the fuzzy-possibilistic co-clustering framework for high-dimensional data analysis. In it, we can develop a clustering algorithm that is simultaneously effective in dealing with high-dimensional data, capable of producing natural and rich representations of clusters, robust to object and feature outliers, and able to intrinsically provide highly-interpretable

descriptions of clusters.

7.2 Future Works

We have discussed some possible future research directions for the works presented in Chapters 4-6. In this section, we go through some potential research directions on a broader level. As we indicate in Chapter 1, there are many issues in clustering that need to be dealt with. Excluding the ones that we study in this thesis, we have issues like parameter settings and scalability. In the conclusions of Chapters 4 and 5, we have mentioned one possible future work is to develop techniques to allow the two fuzzy-possibilistic co-clustering algorithms to automatically decide their optimum parameter settings. These techniques can help us to achieve more consistent and more automated clustering. On scalability, both DFPC and PFCC have the same time complexity of $O(CNK\tau)$, where C denotes the number of clusters, N the number of objects, K the number of features, and τ the number of iterations. Hence, to cluster datasets with very large numbers of objects and features, special techniques, such as sampling or some other extensions, may need to be incorporated into the algorithms.

The formulation of DFPC and PFCC relies on local optimization techniques, and hence they may be prone to local optima problem. Global optimization techniques, such as simulated annealing or genetic algorithm, may be considered for future fuzzy-possibilistic co-clustering algorithms.

Our fuzzy-possibilistic co-clustering framework generates one-level clustering. Some applications, such as web search results clustering, may require clustering outcomes to be represented in hierarchy. Hence, one promising future direction would be to develop hierarchical fuzzy-possibilistic co-clustering algorithms. Not only will this effort help to capture natural taxonomies in datasets, it may also remove the need for users to specify

the number of clusters parameter.

In our current study, we have conducted experiments on artificial and document datasets. However, there are still many other forms of high-dimensional datasets out there, such as gene expression data. Experiments on these other types of high-dimensional dataset may provide additional insights on the performances of the proposed algorithms.

Finally, there are many things that can be further studied on the automatic document clusters summarization application. Some of them have been mentioned in Chapter 6. In addition to these, we may also consider to incorporate some natural language processing (NLP) techniques and other domain-specific information to further improve the quality of the resulting summaries.

Author's Publication List

International Journals

1. W.-C. Tjhi, L. Chen, "Dual fuzzy-possibilistic co-clustering for categorization and summarization of documents," accepted in *IEEE Transaction on Fuzzy Systems*, October 2007.
2. W.-C. Tjhi, L. Chen, "A heuristic-based fuzzy co-clustering algorithm for categorization of high-dimensional data," accepted in *Fuzzy Sets and Systems*, October 2007.
3. W.-C. Tjhi, L. Chen, "Possibilistic fuzzy co-clustering of large document collections," *Pattern Recognition*, vol. 40, no. 12, pp. 3452-3466, 2007.
4. W.-C. Tjhi, L. Chen, "Minimum sum-squared residue for fuzzy co-clustering," *Intelligent Data Analysis*, vol. 10, no. 3, pp. 237-249, 2006.
5. W.-C. Tjhi, L. Chen, "A partitioning-based algorithm to fuzzy co-cluster documents and words," *Pattern Recognition Letters*, vol. 27, no. 3, pp. 151-159, 2006.
6. W.-C. Tjhi, L. Chen, "Improved partitioning-based approach to fuzzy co-clustering," to be submitted.

International Conferences

1. W.-C. Tjhi, L. Chen, “Experimental Study on Fuzzy Word Memberships for Multi-document Summarization,” in *Proceedings of the 6th International Conference on Information, Communications, Signal Processing*, 2007.
2. W.-C. Tjhi, L. Chen, “Robust Fuzzy Co-clustering Algorithm,” in *Proceedings of the 6th International Conference on Information, Communications, Signal Processing*, 2007.
3. W.-C. Tjhi, L. Chen, “Dual Fuzzy-possibilistic Co-clustering and Its Applications for Document Categorization and Summarization,” in *Proceedings of the Optimization-based Data Mining Techniques with Application Workshop, The 7th IEEE International Conference on Data Mining*, 2007.
4. W.-C. Tjhi, L. Chen, “Flexible fuzzy co-clustering with feature-cluster weighting,” in *Proceedings of The 9th International Conference on Control, Automation, Robotics, and Vision*, 2006, pp. 1-6.
5. W.-C. Tjhi and L. Chen, “A new fuzzy co-clustering algorithm for categorization of datasets with overlapping clusters,” in *Lecture Notes in Artificial Intelligence*. Springer, 2006, no. 4093, pp. 328-339.
6. W.-C. Tjhi, L. Chen, “Fuzzy co-clustering of web documents,” in *Proceedings of International Conference on Cyberworld*, 2005, pp. 545-551.

Bibliography

- [1] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park, “Fast algorithms for projected clustering,” in *Proceedings of 1999 ACM SIGMOD International Conference on Management of Data*, 1999, pp. 61–72.
- [2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, “Automatic subspace clustering of high dimensional data for data mining applications,” in *Proceedings of 1998 ACM SIGMOD International Conference on Management of Data*, 1998, pp. 58–65.
- [3] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases,” in *Proceedings of 1993 ACM SIGMOD International Conference on Management of Data*, 1993, pp. 207–216.
- [4] M. Ankerst, M. Breunig, H. P. Kriegel, and J. Sander, “OPTICS: Ordering points to identify the clustering structure,” in *Proceedings of 1999 ACM SIGMOD International Conference on Management of Data*, 1999, pp. 49–60.
- [5] A. Banerjee, I. Dhillon, J. Gosh, S. Merugu, and D. S. Modha, “A generalized maximum entropy approach to Bregman co-clustering and matrix approximation,” in *Proceedings of 2004 International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 509–514.

- [6] D. Barash and D. Comaniciu, “Meanshift clustering for DNA microarray analysis,” in *Proceedings of 2004 IEEE Computational Systems Bioinformatics Conference*, 2004, pp. 578–579.
- [7] M. Barni, V. Cappellini, and A. Mecocci, “Comments on ‘a possibilistic approach to clustering’,” *IEEE Transactions on Fuzzy Systems*, vol. 4, pp. 393–396, 1996.
- [8] G. Beliakov and M. King, “Density based Fuzzy C-means clustering of non-convex patterns,” *European Journal of Operational Research*, vol. 173, pp. 717–728, 2006.
- [9] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum, 1981.
- [10] I. Bolshakov and A. Gelbukh, *Computational Linguistics Models, Resources, Applications*. Instituto Politecnico Nacional, Universidad Nacional Autonoma De Mexico and Fondo De Cultura Economica, 2004.
- [11] C. H. Bong and K. Narayanan, “An empirical study of feature selection for text categorization based on term weightage,” in *Proceedings of 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, 2004, pp. 599–602.
- [12] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” in *Proceedings of the 7th International Conference on World Wide Web*, 1998, pp. 107–117.
- [13] J. R. Brook, P. J. Samson, and S. Sillman, “Aggregation of selected three-day periods to estimate annual and seasonal wet deposition totals for sulfate, nitrate, and acidity. part i: A synoptic and chemical climatology for eastern north america,” *Journal of Applied Meteorology*, vol. 34, no. 2, pp. 297–325, 1995.

- [14] C. Cano, L. Adarve, J. Lopez, and A. Blanco, “Possibilistic approach for biclustering microarray data,” *Computers in Biology and Medicine*, March 2007.
- [15] J. G. Carbonell and J. Goldstein, “The use of MMR, diversity-based reranking for reordering documents and producing summaries,” in *Proceedings of the 21st ACM SIGIR Conference Research and Development in Information Retrieval*, 335–336, Ed., 1998.
- [16] G. Carpenter and S. Grossberg, “A massively parallel architecture for a self-organizing neural pattern recognition machine,” *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 54–115, 1987.
- [17] G. Carpenter, S. Grossberg, and D. Rosen, “Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system,” *Neural Networks*, vol. 4, no. 5, pp. 169–181, 1991.
- [18] D. P. Casasent and X. W. Chen, “New data clustering for RBF classifier of agriculture products from X-ray images,” in *Proceedings of SPIE, Intelligent Robots and Computer Vision XVIII: Algorithms, Techniques, and Active Vision*, D. P. Casasent, Ed., vol. 3837, 1999, pp. 232–241.
- [19] S. Chen, W. Cai, and D. Zhang, “Fast and robust Fuzzy C-means clustering algorithms incorporating local information for image segmentation,” *Pattern Recognition*, vol. 40, no. 3, pp. 825–838, 2007.
- [20] Y. Cheng and G. M. Church, “Biclustering of expression data,” in *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, 2000, pp. 93–103.

- [21] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra, “Minimum sum-squared residues co-clustering of gene expression data,” in *Proceedings of the 4th SIAM International Conference on Data Mining*, 2004, pp. 114–125.
- [22] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 1991.
- [23] M. Cowgill, R. Harvey, and L. Watson, “A genetic algorithm approach to cluster analysis,” *Computers Mathematics with Applications*, vol. 37, pp. 99–108, 1999.
- [24] T. Dasu, D. F. Swayne, and D. Poole, “Grouping multivariate time series: A case study,” in *Proceedings of 2006 KDD Temporal Data Mining Workshop*, 2006, pp. 35–42.
- [25] E. D’Avanzo and B. Magnini, “A keyphrase-based approach to summarization: the LAKE system at DUC-2005,” in *Proceedings of 2005 Document Understanding Conference*, 2005.
- [26] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, “Indexing by Latent Semantic Analysis,” *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391–407, 1990.
- [27] I. S. Dhillon, “Co-clustering documents and words using bipartite spectral graph partitioning,” in *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 269–274.
- [28] I. S. Dhillon, J. Fan, and Y. Guan, “Efficient clustering of very large document collections,” in *Data Mining for Scientific and Engineering Applications*, V. K. R. Grossman, C. Kamath and R. Namburu, Eds. Kluwer Academic Publishers, 2001, invited Book Chapter.

- [29] I. S. Dhillon, S. Mallela, and D. S. Modha, “Information-theoretic co-clustering,” in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003, pp. 89–98.
- [30] C. Ding, T. Li, W. Peng, and H. Park, “Orthogonal nonnegative matrix tri-factorizations for clustering,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 126–135.
- [31] Y. Dong, Y. Zhuang, K. Chen, and X. Tai, “A hierarchical clustering algorithm based on fuzzy graph connectedness,” *Fuzzy Sets and Systems*, vol. 157, no. 13, pp. 1760–1774, 2006.
- [32] R. C. Dubes and A. K. Jain, *Algorithms for Clustering Data*. Prentice Hall, 1988.
- [33] D. Dumitrescu, B. Lazzerini, and L. Jain, *Fuzzy Sets and Their Applications to Clustering and Training*. CRC Press, 2000.
- [34] M. H. Dunham, *Data Mining Introductory and Advanced Topics*. Pearson Education, 2003.
- [35] L. Ertoz, M. Steinbach, and V. Kumar, “A new shared nearest neighbor clustering algorithm and its applications,” in *Proceedings of Workshop on Clustering High Dimensional Data and Its Applications at the 2nd SIAM International Conference on Data Mining*, 2002.
- [36] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases,” in *Proceedings of 1996 International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

- [37] V. Estivill-Castro and I. Lee, “AMOEBa: Hierarchical clustering based on spatial proximity using Delaunay diagram,” in *Proceedings of the 9th International Symposium on Spatial Data Handling*, 1999, pp. 7a.26–7a.41.
- [38] M. Figueiredo and A. K. Jain, “Unsupervised learning of finite mixture models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.
- [39] M. Filippone, F. Masulli, S. Rovetta, S. Mitra, and H. Banka, “Possibilistic approach to biclustering: An application to Oligonucleotide microarray analysis,” in *Proceedings of 2006 Conference on Computational Methods in Systems Biology*, C. Priami, Ed., vol. 4210, 2006, pp. 312–322.
- [40] S. Fisher and B. Roark, “Query-focused summarization by supervised sentence ranking and skewed word distributions,” in *Proceedings of 2006 Document Understanding Conference*, 2006.
- [41] P. Franti, O. Vimajoki, and V. Hautamaki, “Fast agglomerative clustering using a K-nearest neighbor graph,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1875–1881, 2006.
- [42] H. Frigui and O. Nasraoui, “Simultaneous clustering and dynamic keyword weighting for text documents,” in *Survey of Text Mining*, M. W. Berry, Ed. Springer-Verlag, 2004, pp. 45–72.
- [43] —, “Unsupervised learning of prototypes and attribute weights,” *Pattern Recognition*, vol. 37, no. 3, pp. 567–581, 2004.

- [44] B. C. M. Fung, K. Wang, and M. Ester, “Hierarchical document clustering using frequent itemsets,” in *Proceedings of SIAM International Conference on Data Mining*, 2003, pp. 59–70.
- [45] M. K. Ganapathiraju, “Relevance of cluster size in MMR based summarizer,” Self-paced Lab. in Information Retrieval, Tech. Rep., 2002.
- [46] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma, “Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering,” in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2005, pp. 41–50.
- [47] S. Goil, H. Nagesh, and A. Choudhary, “Mafia: Efficient and scalable subspace clustering for very large data sets,” Northwestern University, Tech. Rep., 1999.
- [48] J. Gramm, J. Guo, F. Huffner, and R. Niedermeier, “Graph-modeled data clustering: Fixed-parameter algorithms for clique generation,” *Theory of Computing Systems*, vol. 38, pp. 373–392, 2005.
- [49] S. Guha, R. Rastogi, and K. Shim, “CURE: an efficient clustering algorithm for large databases,” in *ACM SIGMOD International Conference*, 1998, pp. 73–84.
- [50] —, “ROCK: A robust clustering algorithm for categorical attributes,” *Information Systems*, vol. 25, no. 5, pp. 345–366, 2000.
- [51] B. Han, X. Gao, and H. Ji, “Novel feature weighted fuzzy clustering algorithm,” *Journal of Computational Information Systems*, pp. 1141–1146, 2006.
- [52] E. H. Han, G. Karypis, V. Kumar, and M. Bamshad, “Hypergraph based clustering in high-dimensional data sets: A summary of results,” *Data Engineering Bulletin*, no. 1, pp. 15–22, 1998.

- [53] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 1st ed. Morgan Kaufmann, 2000.
- [54] H. Hao, “Fuzzy, evolutionary approaches to image segmentation,” Ph.D. dissertation, Nanyang Technological University, 2002.
- [55] S. Harabagiu and F. Lacatusu, “Topic themes for multi-document summarization,” in *Proceedings of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005, pp. 202–209.
- [56] J. A. Hartigan, “Direct clustering of a data matrix,” *Journal of the American Statistical Association*, vol. 67, no. 337, pp. 123–129, 1972.
- [57] —, *Clustering Algorithms*. John Wiley and Sons, 1975.
- [58] E. Hartuv and R. Shamir, “A clustering algorithm based on graph connectivity,” *Information Processing Letters*, vol. 76, pp. 175–181, 2000.
- [59] R. J. Hathaway and J. C. Bezdek, “Extending fuzzy and probabilistic clustering to very large data sets,” *Computational Statistics and Data Analysis*, vol. 51, no. 1, pp. 215–34, 2006.
- [60] Y. X. He, D. X. Liu, D. H. Ji, H. Yang, and C. Teng, “MSBGA: A multi-document summarization system based on genetic algorithm,” in *Proceedings of the 5th International Conference on Machine Learning and Cybernetics*, 2006, pp. 2659–2664.
- [61] A. Hinneburg and D. A. Keim, “An efficient approach to clustering in large multimedia databases with noise,” in *Proceedings of 1998 International Conference on Knowledge Discovery and Data Mining*, 1998, pp. 58–65.

- [62] J. Jagadeesh, P. Pingali, and V. Varma, “A relevance-based language modeling approach to DUC 2005,” in *Proceedings of 2005 Document Understanding Conference*, 2005.
- [63] A. K. Jain, M. N. Murthy, and P. J. Flynn, “Data clustering: A review,” *ACM Computing Reviews*, pp. 264–323, 1999.
- [64] E. T. Jaynes, “Information theory and statistical mechanics,” *Physical Review*, vol. 106, pp. 620–630, 1957.
- [65] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu, “An efficient K-means clustering algorithm: Analysis and implementation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 881–892, 2002.
- [66] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, “Multilevel hypergraph partitioning: Application in VLSI domain,” in *Proceedings of 1997 ACM/IEEE Design Automation Conference*, 1997, pp. 526–529.
- [67] G. Karypis, E.-H. S. Han, and V. Kumar, “Chameleon: Hierarchical clustering using dynamic modeling,” *Computer*, vol. 32, no. 8, pp. 68–75, 1999.
- [68] L. Kauffman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, 1990.
- [69] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein, “Spectral biclustering of microarray data: Coclustering genes and conditions,” *Genome Research*, vol. 13, no. 4, pp. 703–716, 2003.
- [70] T. Kohonen, *Self-organizing Maps*. Springer-Verlag, 2001.

- [71] I. Kojadinovic, “Agglomerative hierarchical clustering of continuous variables based on mutual information,” *Computational Statistics and Data Analysis*, vol. 46, no. 2, pp. 269–294, 2004.
- [72] R. Krishnapuram and J. M. Keller, “Possibilistic approach to clustering,” *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98–110, 1993.
- [73] —, “The Possibilistic C-means algorithm: Insights and recommendations,” *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 383–395, 1996.
- [74] S. Kullback and R. A. Leibler, “On information and sufficiency,” *Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 1951.
- [75] K. Kummamuru, A. Dhawale, and R. Krishnapuram, “Fuzzy co-clustering of documents and keywords,” in *Proceedings of the 12th IEEE International Conference on Fuzzy Systems*, vol. 2, 2003, pp. 772–777.
- [76] K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram, “A hierarchical monothetic document clustering algorithm for summarization and browsing search results,” in *Proceedings of the 12th International World Wide Web Conference*, 2004, pp. 658–665.
- [77] K. Lagus, S. Kaski, and T. Kohonen, “Mining massive document collections by the Websom method,” *Information Sciences*, vol. 163, pp. 135–156, 2004.
- [78] M. Law, M. Figueiredo, and A. K. Jain, “Simultaneous feature selection and clustering using mixture models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1154–1166, 2004.
- [79] D. J. Lawrie and W. B. Croft, “Generating hierarchical summaries for web searches,” in *Proceedings of the 26th SIGIR*, 2003, pp. 457–458.

- [80] Y. J. Lee, “An automated knowledge extraction system,” Ph.D. dissertation, University of Minnesota, 1994.
- [81] T. Li, S. Ma, and M. Ogihara, “Document clustering via adaptive subspace iteration,” in *The 27th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004, pp. 218–225.
- [82] C. Y. Lin, C. C. Chang, and C. C. Lin, “A new density-based scheme for clustering based on genetic algorithm,” *Fundamenta Informaticae*, vol. 68, no. 4, pp. 315–331, 2005.
- [83] C. Y. Lin and E. H. Hovy, “Automatic evaluation of summaries using N-gram co-occurrence statistics,” in *Proceedings of 2003 Language Technology Conference*, 2003, pp. 71–78.
- [84] C. Y. Lin, *ROUGE Working Note 1.3.1*, Information Sciences Institute University of Southern California, March 2004.
- [85] S. P. Lloyd, “Least squares quantization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, pp. 128–137, 1982.
- [86] B. Long, Z. M. Zhang, and P. S. Yu, “Co-clustering by block value decomposition,” in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2005, pp. 635–640.
- [87] S. Y. Lu and K. S. Fu, “A sentence-to-sentence clustering procedure for pattern analysis,” *IEEE Transactions on Systems Man and Cybernetics*, vol. 8, pp. 381–389, 1978.

- [88] S. C. Madeira and A. L. Oliveira, “Biclustering algorithms for biological data analysis: A survey,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, pp. 24–25, 2004.
- [89] B. Mandhani, S. Joshi, and K. Kummamuru, “A matrix density based algorithm to hierarchically co-cluster documents and words,” in *Proceedings of the 12th International Conference on World Wide Web*, 2003, pp. 511–518.
- [90] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008, ch. Hierarchical Clustering.
- [91] F. Marcelloni, M. Cimino, and B. Lazzerini, “A novel approach to fuzzy clustering based on a dissimilarity relation extracted from data using a TS system,” *Pattern Recognition*, vol. 39, no. 11, pp. 2077–91, 2006.
- [92] F. Masulli and S. Rovetta, “The graded possibilistic clustering model,” in *International Joint Conference on Neural Networks*, 2003, pp. 791–796.
- [93] G. McLachlan and D. Peel, *The EM Algorithm and Extensions*. Wiley, 1997.
- [94] A. A. Melkman and E. Shaham, “Sleeved coclustering,” in *Proceedings of 2004 International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 635–640.
- [95] M. E. S. Mendes and L. Sacks, “Evaluating fuzzy clustering for relevance-based information access,” in *Proceedings of the 12th IEEE International Conference on Fuzzy Systems*, vol. 1, 2003, pp. 648–653.
- [96] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

- [97] S. Miyamoto and M. Mukaidono, “Fuzzy C-means as a regularization and maximum entropy approach,” in *Proceedings of the 7th IFSA World Congress*, 1997, pp. 86–92.
- [98] A. Nenkova and L. Vanderwende, “The impact of frequency on summarization,” Microsoft Research, Tech. Rep., 2005.
- [99] A. Nenkova, L. Vanderwende, and K. McKeown, “A compositional context sensitive multi-document summarizer: Exploring the factors that influence summarizations,” in *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, pp. 573–580.
- [100] R. Ng and J. Han, “CLARANS: A method for clustering objects for spatial data mining,” *IEEE Transactions on Knowledge Data Engineering*, vol. 48, pp. 443–453, 2002.
- [101] C.-H. Oh, K. Honda, and H. Ichihashi, “Fuzzy clustering for categorical multivariate data,” in *Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, vol. 4, 2001, pp. 2154–2159.
- [102] P. Over and J. Yen, “An introduction to DUC 2004: Intrinsic evaluation of generic new text summarization systems,” Document Understanding Conference.
- [103] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, “A Possibilistic Fuzzy C-means clustering algorithm,” *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 4, pp. 517–530, 2005.
- [104] S. Park, J. H. Lee, C. M. Ahn, J. S. Hong, and S. J. Chun, “Query based summarization using non-negative matrix factorization,” in *Proceedings of the 10th*

International Conference on Knowledge-based Intelligent Information and Engineering Systems, B. Gabrys, R. J. Howlett, and L. C. Jain, Eds., 2006, pp. 84–89.

- [105] D. R. Radev, J. Blitzer, A. Winkel, T. Allison, and M. Topper, *MEAD Documentation v3.08*.
- [106] D. R. Radev, H. Jing, M. Stys, and D. Tam, “Centroid-based summarization of multiple documents,” *Information Processing and Management: an International Journal*, vol. 40, pp. 919–938, 2004.
- [107] B.-Y. Ricardo and R.-N. Berthier, *Modern Information Retrieval*. Addison Wesley, 1999.
- [108] K. Rose, E. Gurewitz, and G. C. Fox, “Statistical mechanics and phase transitions in clustering,” *Physical Review Letters*, vol. 65, no. 8, pp. 945–949, 1990.
- [109] P. J. Roussesuw and K. V. Driessen, “A fast algorithm for the minimum covariance determinant estimator,” *Technometrics*, vol. 41, pp. 212–223, 1999.
- [110] E. Ruspini, “A new approach to clustering,” *Information and Control*, vol. 15, pp. 22–32, 1969.
- [111] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information Processing and Management: an International Journal*, vol. 24, no. 5, pp. 513–523, 1998.
- [112] R. Saracoglu, K. Tutuncu, and N. Allahverdi, “A fuzzy clustering approach for finding similar documents using a novel similarity measure,” *Expert Systems with Applications*, vol. 33, no. 3, pp. 600–605, 2007.

- [113] M. Shafiei and E. Milios, “Model-based overlapping co-clustering,” in *Proceedings of the 4th Workshop on Text Mining, The 6th SIAM International Conference on Data Mining*, 2006.
- [114] R. Sharan and R. Shamir, “A clustering algorithm with applications to gene expression analysis,” in *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, 2000, pp. 307–316.
- [115] M. Sinka and D. Corne, “A large benchmark dataset for web document clustering,” in *Soft Computing Systems: Design, Management and Applications, Volume 87 of Frontiers in Artificial Intelligence and Applications*. IOS Pr Inc, 2002, pp. 881–890.
- [116] N. Slonim and N. Tishby, “Agglomerative information bottleneck,” in *Proceedings of the 12th NIPS*, 2000, pp. 617–623.
- [117] M. Steinbach, G. Karypis, and V. Kumar, “A comparison of document clustering techniques,” Department of Computer Science and Engineering, University of Minnesota,, Tech. Rep., 2000.
- [118] A. Strehl, “Relationship-based clustering and cluster ensembles for high-dimensional data mining,” Ph.D. dissertation, The University of Texas at Austin, 2002.
- [119] C. A. Sugar and G. M. James, “Finding the number of clusters in a dataset: An information-theoretic approach,” *Journal of the American Statistical Association*, vol. 98, no. 463, pp. 750–763, 2003.

- [120] A. Tanay, R. Sharan, and R. Shamir, “Biclustering algorithms: A survey,” in *Handbook of Computational Molecular Biology*, A. Srinivas, Ed., vol. 9, 2005, pp. 26.1–26.17.
- [121] H. Timm, C. Borgelt, C. Doring, and R. Kruse, “An extension to possibilistic fuzzy cluster analysis,” *Fuzzy Sets and Systems*, vol. 147, no. 1, pp. 3–16, 2004.
- [122] R. Tishbirani, T. Hastie, M. Eisen, D. Ross, D. Botstein, and P. Brown, “Clustering methods for the analysis of DNA microarray data,” Department of Health Research and Policy, Department of Genetics, and Department of Biochemistry, Stanford University, Tech. Rep., 1999.
- [123] W.-C. Tjhi and L. Chen, “Dual fuzzy-possibilistic co-clustering for categorization and summarization of documents,” accepted in *IEEE Transaction on Fuzzy Systems*, October 2007.
- [124] —, “A heuristic-based fuzzy co-clustering algorithm for categorization of high-dimensional data,” accepted in *Fuzzy Sets and Systems*, October 2007.
- [125] —, “Improved partitioning-based approach to fuzzy co-clustering,” to be Submitted.
- [126] —, “A new fuzzy co-clustering algorithm for categorization of datasets with overlapping clusters,” in *Lecture Notes in Artificial Intelligence*. Springer, 2006, no. 4093, pp. 328–339.
- [127] —, “A partitioning-based algorithm to fuzzy co-cluster documents and words,” *Pattern Recognition Letters*, vol. 27, pp. 151–159, 2006.

- [128] ———, “Possibilistic fuzzy co-clustering of large document collections,” *Pattern Recognition*, vol. 40, no. 12, pp. 3452–3466, 2007, accepted in *Pattern Recognition*, April 2007.
- [129] T. N. Tran, R. Wehrens, and L. M. C. Buydens, “KNN-kernel density-based clustering for high-dimensional multivariate data,” *Computational Statistics and Data Analysis*, vol. 51, pp. 513–525, 2006.
- [130] L. Vanderwende, H. Suzuki, and C. Brockett, “Microsoft research at DUC2006: Task-focused summarization with sentence simplification and lexical expansion,” in *Proceedings of 2006 Document Understanding Conference*, 2006.
- [131] J. J. Verbeek, J. R. J. Nunnink, and N. Vlassis, “Accelerated EM-based clustering of large data sets,” *Data Mining and Knowledge Discovery*, vol. 13, pp. 291–307, 2006.
- [132] P. A. Vijaya, M. N. Murty, and D. K. Subramanian, “Efficient bottom-up hybrid hierarchical clustering techniques for protein sequence classification,” *Pattern Recognition*, vol. 39, no. 12, pp. 2344–2355, 2006.
- [133] K. Wakita and T. Tsurumi, “Finding community structure in mega-scale social networks,” in *International World Wide Web Conference*, 2007, pp. 1275–1276.
- [134] S. S. Weng and M. J. Liu, “Feature-based recommendation for one-to-one marketing,” *Expert Systems with Applications*, vol. 26, no. 4, pp. 493–508, 2004.
- [135] R. Witte and S. Bergler, “Fuzzy coreference resolution for summarization,” in *Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*, Venice, Italy, June 23–24 2003, pp. 43–50.

- [136] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.
- [137] W. Xu and Y. Gong, "Document clustering by concept factorization," in *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004, pp. 202–209.
- [138] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003, pp. 267–273.
- [139] S. Ye and T. S. Chua, "NUS at DUC 2006: Document concept lattice for summarization," in *Proceedings of 2006 Document Understanding Conference*, 2006.
- [140] K. Y. Yeung and W. L. Ruzzo, "Principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.
- [141] J. Yu, "General C-means clustering model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1197–1211, 2005.
- [142] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.
- [143] J.-S. Zhang and Y.-W. Leung, "Improved Possibilistic C-means clustering algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 2, pp. 209–217, 2004.
- [144] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for discovery of association rules," in *Proceedings of 1996 ACM-SIGMOD International Conference on Management of Data*, 1996, pp. 103–114.
- [145] Q. Zhou, L. Sun, and J. Y. Nie, "IS_SUM: A multi-document summarizer based on document index graphic and lexical chains," in *Proceedings of 2006 Document Understanding Conference*, 2006.